

# A Framework for Planning and Controlling Non-Periodic Bipedal Locomotion

Ye Zhao<sup>1</sup>, Benito R. Fernandez<sup>2</sup>, and Luis Sentis<sup>1\*</sup>

<sup>1</sup>*Human Centered Robotics Laboratory, The University of Texas at Austin, TX, USA.*

<sup>2</sup>*Neuro-Engineering Research and Development Laboratory, The University of Texas at Austin, TX, USA.*

## Abstract

This study presents a theoretical framework for planning and controlling agile bipedal locomotion based on robustly tracking a set of non-periodic apex states. Based on the prismatic inverted pendulum model, we formulate a hybrid phase-space planning and control framework which includes the following key components: (1) a step transition solver that enables dynamically tracking non-periodic apex or keyframe states over various types of terrains, (2) a robust hybrid automaton to effectively formulate planning and control algorithms, (3) a phase-space metric to measure distance to the planned locomotion manifolds, and (4) a hybrid control method based on the previous distance metric to produce robust dynamic locomotion under external disturbances. Compared to other locomotion frameworks, we have a larger focus on non-periodic gait generation and robustness metrics to deal with disturbances. Such focus enables the proposed control framework to robustly track non-periodic apex states over various challenging terrains and under external disturbances as illustrated through several simulations. Additionally, it allows a bipedal robot to perform non-periodic bouncing maneuvers over disjointed terrains.

## Keywords

Phase-Space Locomotion Planning, Non-Periodic Apex Stability, Robust Hybrid Automaton, Optimal Control.

## 1 Introduction

Humanoid and legged robots may soon nimbly maneuver over highly rough terrains and cluttered environments. This paper formulates a new framework for the generation of trajectories and an optimal controller to achieve locomotion in those types of environments using a phase-space formalism. Using prismatic inverted pendulum dynamics and given a set of desired apex states, we present a phase-space planner that can precisely negotiate the challenging terrains. The resulting trajectories are formulated as phase-space manifolds. Borrowing from sliding

---

\* Corresponding author; E-mail addresses: yezhao@utexas.edu, benito@austin.utexas.edu, lsentis@austin.utexas.edu.

mode control theory, we use the newly defined manifolds and a Riemannian distance metric to measure deviations due to external disturbances or model uncertainties. A control strategy based on dynamic programming is proposed that steers the locomotion process towards the planned trajectories. Finally, we devise a robust hybrid automaton to effectively formulate control algorithms that involve both continuous and discrete input processes for advanced disturbance recovery.

[Raibert (1986)] pioneered robust hopping locomotion of point-foot monoped and bipedal robots using simple dynamical models but with limited applicability to semi-periodic hopping motions. His focus is on dynamically stabilizing legged robots. Instead, our focus is on precisely tracking apex states, i.e. a discrete set of desired robot center of mass (CoM) positions and velocities along the locomotion paths. Such capability is geared towards the design of highly non-periodic gaits in cluttered environments or the characterization of dynamic gait structure in a generic sense. [Pratt et al. (2001)] achieved point foot biped walking using a virtual model control method but with limited applicability to mechanically supported robots. Unsupported point foot biped locomotion in moderately rough terrains has been recently achieved by [Grizzle et al. (2014)] and [Ramezani et al. (2014)] using Poincaré stability methods. However, Poincaré maps cannot be leveraged to achieving non-periodic gaits for highly irregular or disjointed terrains, which is one of the main applications of our proposed framework.

In [Frazzoli (2001)] a robust hybrid automaton is introduced to achieve time-optimal motion planning of a helicopter in an environment with obstacles. The same group studies robustness to model uncertainties [Schouwenaars et al. (2003)] but ignores external disturbances. More recently, [Majumdar (2013)] accounts for external disturbances (like cross-wind) by computing funnels via Lyapunov functions and switching between these funnels for maneuvering unmanned air vehicles in the presence of obstacles and disturbances. We apply some of these concepts to point-foot locomotion. Our dynamic system is hybrid, i.e., possessing a different set of dynamic equations for each contact stage. As a result, we propose a hybrid control algorithm that switches states when the physical system changes the number of contacts. We use the hybrid automaton framework as a tool for planning and control of bipedal locomotion. We in fact extend their use of hybrid automaton to accommodate for hybrid systems. Additionally we re-generate phase-space trajectories on demand while the previous works rely on pre-generated primitives.

Optimal control of legged locomotion over rough terrains are explored in [Byl and Tedrake (2009); Kuindersma et al. (2015); Dai and Tedrake (2012); Feng et al. (2015)]. [Manchester et al. (2011)] proposed a control technique to stabilize non-periodic motions of under-actuated robots with a focus on walking over uneven terrain. The control is achieved by constructing a lower-dimensional system of coordinates transverse to the target cycle and then computing a receding-horizon feedback controller to exponentially stabilize the linearized dynamics. However, this work does not generate motion trajectories, which is a central part of our work. Also, in contrast with these works, we propose a metric for robustness to recover from disturbances. In [Saglam and Byl (2014)], a controller switching strategy for walking on irregular terrains is proposed. They optimize policies for switching between a set of known controllers. Their method is further extended to incorporate noise on the terrain and through a value iteration process they achieve a certain degree of robustness through switching. However, they focus on mesh learning by considering a set of known controllers. Instead, our paper is focused on creating new controllers from scratch for general types of terrains. Additionally, their work is focused on 2D locomotion whereas we focus on 3D.

A general optimal motion control framework for behavior synthesis of human-like avatars is presented in [Mordatch et al. (2012)]. One key missing aspect is quantifying robustness and analyzing feedback stability. Additionally, this work does not address locomotion of point foot robots.

In light of this discussion, our contributions are the following: (1) we formulate a hybrid automaton to characterize non-periodic locomotion dynamics, (2) using the automaton, we synthesize motion plans in the phase-space to maneuver over irregular terrains while tracking a set of desired keyframes (i.e. apex states), (3) a phase-space manifold is created with a Riemannian distance metric to measure nominal trajectory deviations and design an in-step controller, and (4) we derive an optimal control framework to recover from disturbances and uncertainty and study its stability. Overall, the key difference with previous works is our focus on trajectory generation and robust control of non-periodic gaits. We are less centered on dynamic balance or moving from a start to a finish location but instead on tracking desired keyframes composed of discrete robot CoM positions and velocities. As such our framework provides a greater level of granularity that makes it more suitable for designing gates in cluttered environments.

## 2 Related Work

The Capture Point method [Pratt et al. (2006)] provides one of the most practical frameworks for locomotion. Sharing similar core ideas, divergent component of motion [Takenaka et al. (2009)] and extrapolated center of mass [Hof (2008)] were independently proposed. Extensions to the Capture Point method, [Morisawa et al. (2012); Engelsberger et al. (2015)], allow locomotion over rough terrains. Motion planning techniques based on interpolation through kinematic configurations have been explored, among other works, by [Hauser (2014)] and [Pham et al. (2013)]. Those techniques are making great progress towards mobility and locomotion in various kinds of environments. The main difference from these studies is that our controller explicitly accounts for robustness and stability to achieve under-actuated dynamic walking. If these works were to be implemented in unstable robots such as point-foot bipeds, they would lose balance and fail to recover. Our planner allows for continuous recovery without explicitly controlling the robot's center of mass.

In [Hobbelen and Wisse (2007)], a gait sensitivity norm is presented to measure disturbance rejection during dynamic walking. In [Hamed et al. (2015)], sensitivity analysis with respect to ground height variations is performed to model robustness of orbits. These techniques are limited to cyclic walking gaits. The work in [Arslan and Saranli (2012)] unifies planning and control to provide robustness. In [Nguyen and Sreenath (2015)], an optimal robust controller is designed based on control Lyapunov function to achieve bipedal locomotion with model uncertainties. However, the techniques in these two works are only applied to planar robots.

Numerous studies have focused on recovery strategies upon disturbances [Hofmann (2006); Zhao et al. (2013)]. Various recovery methods have been proposed based on ankle, hip, knee, and stepping strategies [Kuo and Zajac (1992); Stephens (2007)]. In [Hyon and Cheng (2007)], a stepping controller based on ground contact forces is implemented in a humanoid robot. The study in [Komura et al. (2005)] controlship angular momentum to achieve planar bipedal locomotion. In our study, we simultaneously control angular momentum, CoM height and foot placements to achieve unsupported rough terrain walking.

Online trajectory optimization is explored in [Tassa et al. (2012); Audren et al. (2014)] for complex humanoid behaviors. [Stephens and Atkeson (2010)] uses model predictive control (MPC) for push recovery by planning future steps. [Wieber (2006)] presents a linear MPC scheme for zero moment point control with perturbations. The problem of MPC is that it cannot find a globally optimal solution due to the finite horizon. In contrast, we use

dynamic programming to exhaustively search for the global optimal solution. Since the inverted pendulum model has low dimensional states, the “curse of dimensionality” is not an issue for our case.

### 3 Problem Definition

We first present basic control formalism and manifold analysis that will allow us to characterize, plan and control non-periodic locomotion processes in later sections.

#### 3.1 System Equations

Legged robots can be characterized as Multi-Input/Multi-Output (MIMO) systems. Let us assume that a bipedal robot can be characterized by  $n_j$  joint degrees of freedom (DOF),  $\mathbf{q} = [q_1, q_2, \dots, q_{n_j}]^T \in \mathbb{R}^{n_j}$ . Letting  $\mathbf{x}(t) = [\mathbf{q}^T(t), \dot{\mathbf{q}}^T(t)]^T \in \mathbb{R}^n$ , be the state-space vector ( $n = 2n_j$ ),  $\mathbf{u}(t) \in \mathbb{R}^m$ , represents the control input vector (generalized torques and forces), and defining  $\mathbf{f}(\mathbf{x}(t))$ ,  $\mathbf{g}(\mathbf{x}(t))$ , and  $\mathbf{h}(\mathbf{x}(t))$  in the obvious manner, the mechanical model is expressed in state variable form as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{J}_d(\mathbf{x}(t))\mathbf{d}(t), \quad (1a)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)), \quad (1b)$$

where,  $\mathbf{d}(t)$  represent the generalized external disturbance forces, and  $\mathbf{J}_d(\mathbf{q}(t))$  is the disturbance distribution matrix. The output vector,  $\mathbf{y}(t) = [y_1, y_2, \dots, y_p]^T \in \mathbb{R}^p$  is generated by  $\mathbf{h}(\mathbf{x}(t))$ , that may represent positions and/or velocities in the task space. Without loss of generality, let us consider systems in the normal form, where  $\mathbf{h}(\cdot)$  is at least  $\mathcal{C}^r$ , where  $r$  is the relative order of the output. The disturbances and modeling errors satisfy the matching conditions [Fernández-Rodríguez (1988)].

#### 3.2 System Normalization for Phase-Space Control Design

General robotic systems are not in normal form, but we can transform them by finding what relative order of the output derivatives are explicitly controllable. Each of the outputs  $y_i$  in Eq. (1) has a relative order  $r_i$ , defined by the smallest derivative order where the control appears,

$$y_i^{[k]} = \frac{d^k y_i}{dt^k} = \mathcal{L}_{\mathbf{f}}^k(h_i(\mathbf{x})) + \mathcal{L}_{\mathbf{g}}(\mathcal{L}_{\mathbf{f}}^{k-1}(h_i(\mathbf{x})))\mathbf{u}, \quad (2a)$$

$$\mathcal{L}_{\mathbf{g}}(\mathcal{L}_{\mathbf{f}}^{k-1}(h_i(\mathbf{x}))) = 0 \quad \text{for } 0 \leq k < r_i, \quad (2b)$$

$$y_i^{[r_i]} = \mathcal{L}_{\mathbf{f}}^{r_i}(h_i(\mathbf{x})) + \mathcal{L}_{\mathbf{g}}(\mathcal{L}_{\mathbf{f}}^{r_i-1}(h_i(\mathbf{x})))\mathbf{u}, \quad (2c)$$

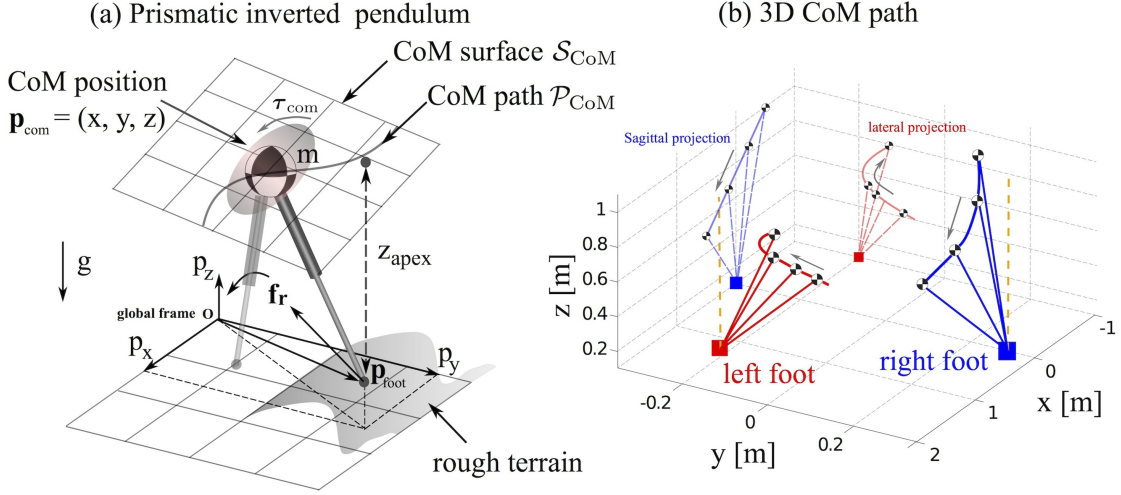
$$\mathcal{L}_{\mathbf{g}}(\mathcal{L}_{\mathbf{f}}^{r_i-1}(h_i(\mathbf{x}))) \neq 0 \quad \forall \mathbf{x} \in \mathbb{S}_i \subset \mathbb{R}^n, \quad (2d)$$

where,  $\mathcal{L}_{\mathbf{f}}^0(h_i(\mathbf{x})) = h_i(\mathbf{x})$ ,  $\mathcal{L}_{\mathbf{f}}(\mathbf{h})$  and  $\mathcal{L}_{\mathbf{g}}(\mathbf{h})$  are the directional Lie derivatives of function  $\mathbf{h}(\mathbf{x})$  in the directions of  $\mathbf{f}(\mathbf{x})$  and  $\mathbf{g}(\mathbf{x})$  respectively [Isidori (1985)], and  $\mathbb{S}_i$  is the output-controllable subspace, where the Lie derivative in Eq. (2) does not vanish,

$$\mathbb{S}_i = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathcal{L}_{\mathbf{g}}(\mathcal{L}_{\mathbf{f}}^{r_i-1}(h_i(\mathbf{x}))) \neq 0 \right\}. \quad (3)$$

The relative order tells us that the  $r_i^{\text{th}}$ -derivative of output  $y_i$  can be explicitly controlled. The region where  $\mathbb{S}_i$  vanishes, entails the system loses relative order and hence the  $r_i^{\text{th}}$ -derivative is no longer controllable (at least





**Fig. 1.** 3D prismatic inverted pendulum model. (a) We define a prismatic inverted pendulum model with all of its mass located at its base while equipping it with a flywheel to generate moments. We restrict the movement of the center of mass to 3D planes (surfaces)  $\mathcal{S}_{\text{CoM}}$ . (b) shows motions of pendulum dynamics restricted to a 3D plane.

explicitly). For a controllable system,  $r_i \leq n$ . Following the normalization procedure, we get the output controllable subspace,

$$\xi_{i,1} = y_i = h_i(\mathbf{x}) = \mathcal{L}_f^0(h_i(\mathbf{x})), \quad (4a)$$

$$\xi_{i,j} = y_i^{[j-1]} = \dot{\xi}_{i,j-1} = \mathcal{L}_f^{j-1}(h_i(\mathbf{x})) \quad \text{for } 1 < j < r_i, \quad (4b)$$

$$\xi_{i,r_i} = y_i^{[r_i]} = \dot{\xi}_{i,r_i} = \mathcal{L}_f^{r_i}(h_i(\mathbf{x})) + \mathcal{L}_g(\mathcal{L}_f^{r_i-1}(h_i(\mathbf{x})))\mathbf{u}(t). \quad (4c)$$

The output space variables,  $\xi_i = [\xi_{i,1}, \xi_{i,2}, \dots, \xi_{i,r_i-1}]^T \in \mathbb{R}^{r_i}$  represent the phase-space for the  $i$ -th output. For instance, the output phase-space for locomotion control could be chosen to be the robot's center of mass. We can concatenate all  $\xi_i$ ,  $\forall i = 1, 2, \dots, m$  into a single phase-space vector  $\xi = [\xi_1^T, \xi_2^T, \dots, \xi_m^T]^T \in \mathbb{R}^r$ , where,  $r = \sum r_i$ . For phase-space motion, we define a phase-space manifold  $\mathcal{M}_i$  for each task-space output,  $y_i$ , in terms of its phase-space vector,  $\xi_i$ ,

$$\mathcal{M}_i = \left\{ \xi_i \in \mathbb{R}^{r_i} \subset \mathbb{R}^n \mid \sigma_i \triangleq \sigma_i(\xi_i) = 0 \right\}, \quad (5)$$

where,  $\sigma_i$  is referred to as the  $i^{\text{th}}$  element of deviation vector, which measures the deviation distance from the manifold  $\mathcal{M}_i$  using a Riemannian metric. In order to be able to control this deviation, the order of the manifold is one less than the relative order of the  $i^{\text{th}}$ -output, i.e.,  $r_i - 1$ . For most legged robots (not considering actuator dynamics), the relative order is  $r = 2$ .

## 4 Prismatic Inverted Pendulum Dynamics on a Parametric Surface

The dynamics of point foot bipedal robots in generic terrain topologies during single contact can be mechanically approximated as an inverted pendulum model (see Fig. 1). Our model consists of a prismatic massless joint with all the mass concentrated on the hip position, defined as the 3D CoM position,  $\mathbf{p}_{\text{com}} = (x, y, z)^T$ , and a flywheel

spinning around it, with orientation angles  $\mathbf{R} = (\phi, \theta, \psi)^T$ . The objective of locomotion is to move the robot's CoM along a certain path from point A to B over a terrain. As such, we first specify a 3D surface,  $\mathcal{S}_{\text{CoM}}$ , where the CoM path will exist, which in general, may have the following implicit form,

$$\mathcal{S}_{\text{CoM}} = \left\{ \mathbf{p}_{\text{com}} \in \mathbb{R}^3 \mid \psi_{\text{CoM}}(\mathbf{p}_{\text{com}}) = 0 \right\}. \quad (6)$$

This surface can be specified in various ways, such as via piecewise arc geometries [Mordatch et al. (2010); Srinivasan and Ruina (2006)]. Once the controller is designed, the CoM will follow a concrete trajectory  $\mathcal{P}_{\text{CoM}}$  (as shown in Fig. 1), which we specify via piecewise splines described by a progression variable,  $\zeta \in [\zeta_{j-1}, \zeta_j]$ , for the  $j^{\text{th}}$  path manifold, i.e.

$$\mathcal{P}_{\text{CoM}} = \bigcup_j \mathcal{P}_{\text{CoM}_j} \subseteq \mathcal{S}_{\text{CoM}}, \quad \mathcal{P}_{\text{CoM}_j} = \left\{ \mathbf{p}_{\text{com}_j} \in \mathbb{R}^3 \mid \mathbf{p}_{\text{com}_j} = \sum_{k=0}^{n_p} \mathbf{a}_{jk} \zeta^k \right\}, \quad (7)$$

where  $n_p$  is the order of the spline degree. The progression variable  $\zeta$  is therefore the arc length along the CoM path acting as the Riemannian metric for distance. Each  $\mathbf{a}_{jk} \in \mathbb{R}^3$  is the coefficient vector of  $k^{\text{th}}$  order. To guarantee the spline smoothness,  $\mathbf{p}_{\text{com}}$  requires the connection points, i.e. the knots at progression instant  $\zeta_j$ , to be  $\mathcal{C}^{n_p-1}$  continuous,

$$\mathbf{p}_{\text{com}_j}^{[l]}(\zeta_j) = \frac{d^l \mathbf{p}_{\text{com}_j}}{d\zeta^l}(\zeta_j) = \mathbf{p}_{\text{com}_{j+1}}^{[l]}(\zeta_j), \quad \forall 0 \leq l \leq n_p - 1. \quad (8)$$

The purpose of introducing the CoM manifold  $\mathcal{S}_{\text{CoM}}$  is to constrain CoM motions on surfaces that are designed to conform to generic terrains while allowing free motion within this surface. Following a concrete CoM path is achieved by selecting proper control inputs as we will see further down. The CoM path manifold,  $\mathcal{P}_{\text{CoM}}$  (embedded in  $\mathcal{S}_{\text{CoM}}$ ), can be represented in the phase-space,  $\boldsymbol{\xi}$ . We call this representation as the *phase-space manifold* and define it as,

$$\mathcal{M}_{\text{CoM}} = \bigcup_j \mathcal{M}_{\text{CoM}_j}, \quad \mathcal{M}_{\text{CoM}_j} = \left\{ \boldsymbol{\xi} \in \mathbb{R}^6 \mid \sigma_j(\boldsymbol{\xi}) = 0 \right\}, \quad (9)$$

which is the core manifold used in our phase-space planning and control framework. The function  $\sigma_j(\boldsymbol{\xi})$  is an implicit function in the phase space measuring the distance to the manifold.

#### 4.1 Dynamic Equations of Motion

Besides the CoM path surface previously described, pendulum dynamics can be characterize via formulating the dynamic balance of moments of the pendulum system. For our single contact scenario, the sum of moments,  $\mathbf{m}_i$ , with respect to the global reference frame (see Fig. 1) is

$$\sum_i \mathbf{m}_i = -\mathbf{p}_{\text{foot}} \times \mathbf{f}_r + \mathbf{p}_{\text{com}} \times (\mathbf{f}_{\text{com}} + m\mathbf{g}) + \boldsymbol{\tau}_{\text{com}} = 0, \quad (10)$$

where,  $\mathbf{p}_{\text{foot}} = (x_{\text{foot}}, y_{\text{foot}}, z_{\text{foot}})^T$  is the position of the foot contact point (i.e., left or right foot),  $\mathbf{f}_r$  is the three dimensional vector of ground reaction forces,  $\mathbf{f}_{\text{com}} = m(\ddot{x}, \ddot{y}, \ddot{z})^T$  is the vector of center of mass inertial forces,

$\boldsymbol{\tau}_{\text{com}} = (\tau_x, \tau_y, \tau_z)^T$  is the vector of angular moments of the modeled flywheel attached to the inverted pendulum,  $m$  is the total mass, and  $\mathbf{g} \in \mathbb{R}^3$  corresponds to the gravity field. The system's linear force equilibrium can be formulated as  $\mathbf{f}_r = \mathbf{f}_{\text{com}} + m\mathbf{g}$ , allowing us to simplified Eq. (10) to:

$$(\mathbf{p}_{\text{com}} - \mathbf{p}_{\text{foot}}) \times (\mathbf{f}_{\text{com}} + m\mathbf{g}) = -\boldsymbol{\tau}_{\text{com}}. \quad (11)$$

For our purposes, we consider only the class of prismatic inverted pendulums whose center of mass is restricted to a path surface  $\mathcal{S}_{\text{CoM}}$  as indicated in Eq. (6). Moreover, for simplicity we only consider 3D surfaces that are invariant to the lateral CoM coordinate as we will see in Eq. (14). This consideration allows to decouple the dynamics of the Sagittal plane from those of the lateral plane. Considering as our output state the CoM positions,  $\mathbf{p}_{\text{com}}$ , the state space,  $\boldsymbol{\xi} = (\mathbf{p}_{\text{com}}^T, \dot{\mathbf{p}}_{\text{com}}^T)^T = (x, y, z, \dot{x}, \dot{y}, \dot{z})^T \in \Xi \subseteq \mathbb{R}^6$  is the phase-space vector. Then from Eq. (11) it can be shown that the prismatic inverted pendulum model for a  $q^{\text{th}}$  walking step, is simplified to the following control system

$$\dot{\boldsymbol{\xi}} = \mathcal{F}(q, \boldsymbol{\xi}, \mathbf{u}) = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_q^2(x - x_{\text{foot}_q}) - \frac{\omega_q^2}{mg}\tau_y \\ \omega_q^2(y - y_{\text{foot}_q}) - \frac{\omega_q^2}{mg}\tau_x \\ a_q\omega_q^2(z - z_{\text{foot}_q}) - \frac{a_q\omega_q^2}{mg}\tau_y \end{pmatrix}, \quad (12)$$

where the phase-space asymptotic slope is defined as

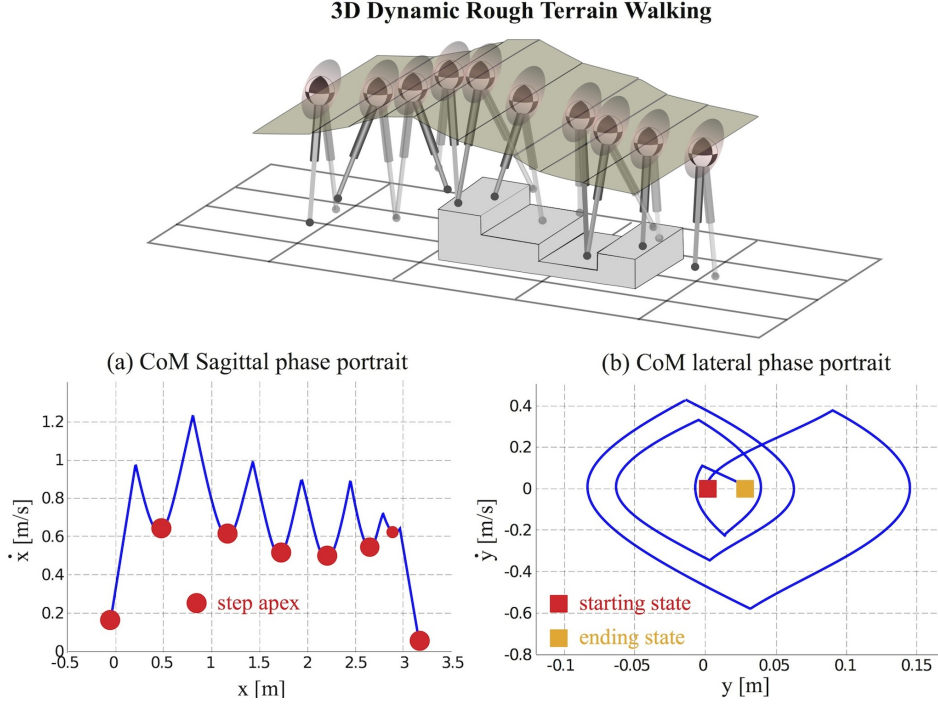
$$\omega_q = \sqrt{\frac{g}{z_{\text{apex}_q}}}, \text{ with } z_{\text{apex}_q} = (a_q \cdot x_{\text{foot}_q} + b_q - z_{\text{foot}_q}), \quad (13)$$

where  $g$  is the gravity constant,  $a_q$  and  $b_q$  are the slope and constant scalars for the linear CoM path surfaces that we consider (see Fig. 2), i.e.

$$\mathcal{S}_{\text{CoM}_q} = \left\{ (x, y, z) \in \mathbb{R}^3 \mid \psi_{\text{CoM}_q}(x, y, z) = z - a_q x - b_q = 0 \right\}. \quad (14)$$

We have defined  $z_{\text{apex}_q}$  such that it corresponds to the vertical distance between the CoM and the location of the foot contact at the instant where their Sagittal projections are identical, i.e. the *Sagittal Apex*;  $\mathcal{F}$  represents a vector field of inverted pendulum dynamics. In general, there will be a hybrid control policy,  $\mathbf{u} = \boldsymbol{\pi}(q, \boldsymbol{\xi})$ , defined by the control variables  $\mathbf{u} = \{\omega_q, \boldsymbol{\tau}_{\text{com}_q}, \mathbf{p}_{\text{foot}_q}\} \in \mathcal{U}$ , where,  $\mathcal{U}$  is an open set of admissible control values. The sets  $\Xi$  and  $\mathcal{U}$  are assumed to be compact.

**Definition 1** (Sagittal Apex). *The Sagittal apex occurs when the projection of the CoM is equal to the location of the foot contact in the system's Sagittal axis.*



**Fig. 2.** 3D phase-space planning. Given step apex conditions, single contact dynamics generate the valley profiles shown in (a). (b) depicts a similar strategy in the lateral plane. However, since foot transitions have already been determined, what is left is to determine foot lateral positions. This is done so the lateral CoM behavior shown in (b) follows a semi-periodic trajectory that is bounded within a closed region.

#### 4.2 Nominal Phase-Space Trajectory Generation

We will first focus on the generation of trajectories in the Sagittal plane of the robot's walking reference. Sagittal dynamics are represented - ignoring for simplicity, the discrete variable,  $q$ , - in the first and fourth row of the system of Eq. (12), i.e.

$$\dot{\mathbf{x}} = \mathcal{F}_{\mathbf{x}}(\mathbf{x}, \mathbf{u}_{\mathbf{x}}) = \begin{pmatrix} \dot{x} \\ \omega^2(x - x_{\text{foot}}) - \frac{\omega^2}{mg}\tau_y \end{pmatrix}. \quad (15)$$

This system would be fully controllable if its continuous control inputs,  $\omega$  and  $\tau_y$ , were unconstrained. However, their limited range urges us to first consider the motion trajectories under nominal (i.e. open loop) values. As we previously motivated in Eq. (14), the path manifold,  $\mathcal{S}_{\text{CoM}}$  is defined a priori to conform to the terrains via methods not considered in this paper. From Eq. (13), once the path manifold is defined and for known contact locations, the set of phase-space asymptotic slopes,  $\omega$ , is also known as shown in Eq. (13). For simplicity, the nominal flywheel moments are designed to be null, i.e.  $\tau_y = 0$ . Under these considerations, the following algorithm produces nominal phase-space trajectories of the system's center of mass in the Sagittal direction of reference:

**Algorithm 1.** Nominal Phase-Space Trajectory Generation.

**Input:**

- (i):  $\mathcal{S}_{\text{CoM}} \leftarrow \{\mathcal{S}_{\text{CoM}_q} : [\zeta_{q-1}, \zeta_q] \rightarrow \mathbb{R}^3, \forall q = 1, \dots, N\}$
- (ii):  $x_{\text{foot}} \leftarrow \{x_{\text{foot}_1}, x_{\text{foot}_2}, \dots, x_{\text{foot}_N}\}$

(iii):  $\dot{x}_{\text{apex}} \leftarrow \{\dot{x}_{\text{apex}_1}, \dot{x}_{\text{apex}_2}, \dots, \dot{x}_{\text{apex}_N}\}$

(iv):  $\tau_y(t) \leftarrow 0$

**Operation:**

(i):  $\omega := \{\omega_1, \omega_2, \dots, \omega_N\} \leftarrow \mathcal{S}_{\text{CoM}}$

(ii):  $\ddot{x}(t) \leftarrow \text{PIPM}(\omega, \tau_y(t), x_{\text{foot}})$

(iii):  $(x_{k+1}, \dot{x}_{k+1}) \leftarrow \text{ODE}(x_k, \dot{x}_k, \ddot{x}_k, \epsilon)$

**Output:**

Phase-space trajectories  $\mathcal{M}_{\text{CoM}} := \bigcup_q \mathcal{M}_{\text{CoM}_q}$

A similar algorithm could be written to generate motion in the lateral direction. Here,  $\dot{x}_{\text{apex}}$ , represents desired apex velocities, PIPM represents the prismatic inverted pendulum model on the parametric surfaces outlined in Eq. (12), ODE represents the numerical integration of the ordinary differential equation associated with the model, with numerical step  $\epsilon$ , and  $k$  represents the discretization of the output state for numerical integration purposes. Trajectories for multiple steps of a locomotion sequence on rough terrain are simulated using this process in Fig. 2.

## 5 Hybrid Phase-Space Motion Planning

In this section we propose a hybrid robust automaton [Branicky et al. (1998); Frazzoli (2001); Lygeros et al. (2008)] with the following key features: I) an invariant set and a recoverability set to characterize control robustness, i.e., the bundle of attractiveness, and II) a non-periodic step transition strategy based on the previously described phase-space trajectories. The hybrid automaton governs the planner's behavior across multiple walking steps and as such constitutes the theoretical core of our proposed phase-space locomotion planning framework.

We continue our focus on Sagittal plane dynamics first, then extend the planner to all directions. For practical purposes we will use the symbol  $\mathbf{x} = \{x, \dot{x}\}$  to describe the Sagittal state space associated with CoM dynamics. Note that this symbol represents now the output dynamics outlined in Eq. (4) instead of the robot plant of Eq. (1). Eq. (9) can thus be re-considered in the output space as  $\mathcal{M}_{\text{CoM}_q} = \{\mathbf{x} \in \mathcal{X} \mid \sigma_q(\mathbf{x}) = 0\}$  where  $\sigma_q$  (the normal distance) representing the deviation from the manifold  $\mathcal{M}_{\text{CoM}_q}$ .

**Definition 2** (Invariant Bundle). *A set  $\mathcal{B}_q(\epsilon)$  is an invariant bundle if, given  $\mathbf{x}_{\zeta_0} \in \mathcal{B}_q(\epsilon)$ , with  $\zeta_0 \in \mathbb{R}_{\geq 0}$ , and increment  $\epsilon > 0$ ,  $\mathbf{x}_\zeta$  stays within an  $\epsilon$ -bounded region of  $\mathcal{M}_{\text{CoM}_q}$ ,*

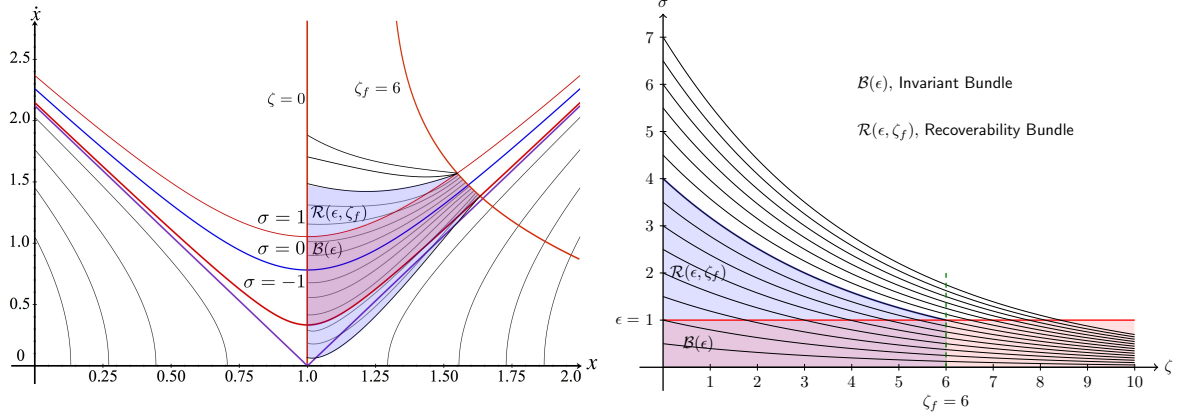
$$\mathcal{B}_q(\epsilon) = \left\{ \mathbf{x} \in \mathcal{X} \mid |\sigma_q(\mathbf{x})| \leq \epsilon \right\}, \quad (16)$$

where,  $\zeta_0$  and  $\zeta$  are initial and current phase progression variables, respectively.  $\mathbf{x}_{\zeta_0}$  is an initial condition.

This type of bundle characterizes “robust subspaces” (i.e., “tubes”) around nominal phase-space trajectories which guarantee that, if the state initializes within this space, it will remain on it.

**Definition 3** (Finite-Phase Recoverability Bundle). *The invariant bundle  $\mathcal{B}_q(\epsilon)$  around a phase-space manifold  $\mathcal{M}_{\text{CoM}_q}$  has a finite phase recoverability bundle,  $\mathcal{R}_q(\epsilon, \zeta_f) \subseteq \mathcal{X}$  defined as,*

$$\mathcal{R}_q(\epsilon, \zeta_f) = \left\{ \mathbf{x}_\zeta \in \mathcal{X}, \quad \zeta_0 \leq \zeta \leq \zeta_f \mid \mathbf{x}_{\zeta_f} \in \mathcal{B}_q(\epsilon) \right\}. \quad (17)$$



**Fig. 3.** Mapping between Cartesian phase-space and  $\zeta - \sigma$  space. The two subfigures show the invariant bundle,  $B(\epsilon)$  (shown in red) and the recoverability bundle,  $R(\epsilon, \zeta_f)$  (shown in blue) in different spaces. The left subfigure shows Cartesian phase-space while the right one shows  $\zeta - \sigma$  space ( $\sigma$  denotes the phase-space manifold as defined in Eq. (9)). The figure on the right only shows positive bundles of  $\sigma$  while the negative ones are symmetric about the  $\zeta$  axis. Since this is a Euclidean space, the manifold for a constant  $\sigma$  is a horizontal line and constant values of  $\zeta$  are vertical lines. If the condition when we expect the transition to occur is at  $\zeta = \zeta_f$ , the recoverability bundle shows the range of perturbations that can be tolerated at different  $\zeta$  – the system recovers to the invariant bundle before  $\zeta_f$ .

Note that this bundle assumes the existence of a control policy for recoverability. We will later use these metrics to characterize robustness of our controllers. Visualization of the invariant and recoverability bundles are shown in Fig. 3.

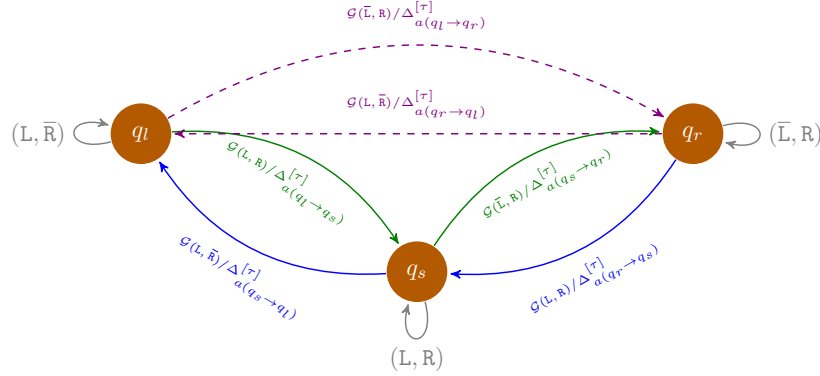
### 5.1 Hybrid Locomotion Automaton

Legged locomotion is a naturally hybrid control system, with both continuous and discrete dynamics. We define discrete states  $\mathcal{Q} = \{q_l, q_r, q_s\}$  representing the support of the left foot  $q_l$  or the right foot  $q_r$  or both  $q_s$  (stance) feet as shown in Figs. 2 and 4. On each phase, the continuous dynamics are represented as shown in Eq. (15) and over a domain  $\mathcal{D}(q)$ . We characterize the hybrid system as a directed graph  $(\mathcal{Q}, \mathcal{E})$  (see Fig. 4), with nodes represented by  $q \in \mathcal{Q}$  and edges represented by  $\mathcal{E}(q, q + 1)$ , that characterize the transitions between nodes. The transitions between states can be grouped into eight classes depending on whether a vector field or variable changes discontinuously and what the trigger mechanism is. Table 1 shows the transition classification.

Table 1: Transition Classifications. System vector field is  $\mathcal{F}$  as shown in Eq. (12).

Type	Transition	Switching	Jump
Autonomous	$\Delta_a^{[\tau]}$	$\mathcal{F}_x^+(\cdot, \cdot, \cdot, x^+, \cdot, \cdot) \leftarrow \Delta_a^{[\delta_s]}(x^-)$	$x^+ \leftarrow \Delta_a^{[\delta_j]}(x^-)$
Controlled	$\Delta_c^{[\tau]}$	$\mathcal{F}_x^+(\cdot, \cdot, \cdot, u_x^+, \cdot) \leftarrow \Delta_c^{[\delta_s]}(u_x^-)$	$u_x^+ \leftarrow \Delta_c^{[\delta_j]}(u_x^-)$
“Timed”	$\Delta_t^{[\tau]}$	$\mathcal{F}_x^+(\zeta, \cdot, \cdot, \cdot, \cdot, \cdot) \leftarrow \Delta_d^{[\delta_t]}(\zeta)$	$x^+ \leftarrow \Delta_t^{[\delta_j]}(\zeta)$
“Disturbed”	$\Delta_d^{[\tau]}$	$\mathcal{F}_x^+(\cdot, \cdot, \cdot, \cdot, w_d) \leftarrow \Delta_d^{[\delta_s]}(w_d)$	$x^+ \leftarrow \Delta_d^{[\delta_j]}(w_d)$

The hybrid automaton state is given by:  $s = (\zeta, q, x^T)^T$ .  $\tau \in \{\delta_s, \delta_j\}$  represents the “switching” or “jump” transition types respectively. Detailed definitions for these transitions can be found in [Branicky et al. (1998)]. The



**Fig. 4.** This figure shows the hybrid locomotion automaton for a walking biped robot. This automaton has three generic continuous modes,  $\mathcal{Q} = \{q_l, q_s, q_r\}$ , that represent when the robot is standing in the left leg only ( $q_l$ ), standing in the right leg only ( $q_r$ ), and when the robot is standing in both legs ( $q_s$ ). Shown in the edges are the condition for the transition, the guard  $\mathcal{G}(q, q+1)$  and effect of the transition in the state and outputs, the map  $\Delta_\mu^{[\tau]}$ . Note that, each mode is non-periodic.

condition that triggers the type of event (switching or jump) is determined by a guard  $\mathcal{G}(q, q+1)$  for the particular edge  $\mathcal{E}_{q, q+1}$ . With this information, let us formulate a robust hybrid automaton for our locomotion planning.

**Definition 4.** A phase-space robust hybrid automaton is a dynamical system, described by a  $n$ -tuple

$$\text{PSRHA} := (\zeta, \mathcal{Q}, \mathcal{X}, \mathcal{U}, \mathcal{W}, \mathcal{F}, \mathcal{I}, \mathcal{D}, \mathcal{R}, \mathcal{B}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \Delta), \quad (18)$$

where,  $\zeta$  is the phase-space progression variable,  $\mathcal{Q}$  is the set of discrete states,  $\mathcal{X}$  is the set of continuous states,  $\mathcal{U}$  is the set of control inputs,  $\mathcal{W}$  is the set of disturbances,  $\mathcal{F}$  is the vector field,  $\mathcal{I}$  is the initial condition,  $\mathcal{D}$  is the domain,  $\mathcal{R}$  is the collection of recoverability sets,  $\mathcal{B}$  is the collection of invariant bundles,  $\mathcal{E} := \mathcal{Q} \times \mathcal{Q}$  is the edge,  $\mathcal{G} : \mathcal{Q} \times \mathcal{Q} \rightarrow 2^{\mathcal{X}}$  is the guard,  $\mathcal{T}$  is the transition termination set, and  $\Delta$  is the transition map.

## 5.2 Step Transition Strategy

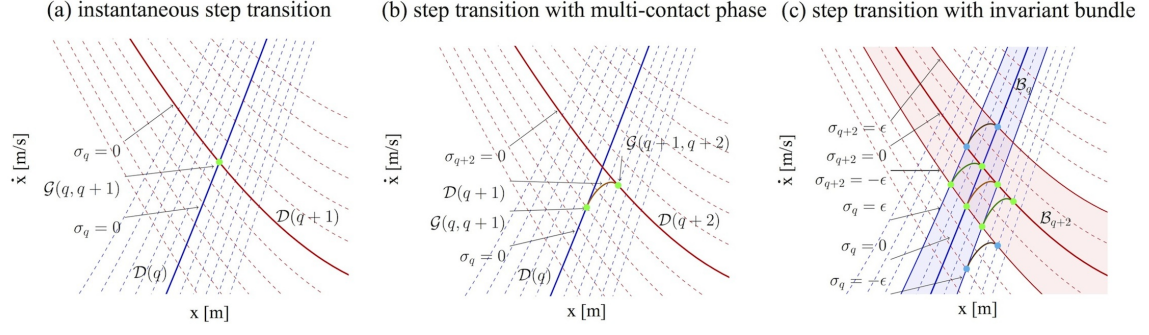
Step transitions could be idealized as an instantaneous contact (Fig. 5 (a)) or have a short dual-contact phase (Fig. 5(b)). We first create a strategy for instantaneous contact switch then extend it to multi-contact in Appendix D.

**Definition 5** (A Phase-Space Simple Walking Step). A  $q^{\text{th}}$  simple walking step is defined as a phase-space trajectory in domain  $\mathcal{D}(q)$ , which has two boundary guards  $\mathcal{G}(q-1, q)$  and  $\mathcal{G}(q, q+1)$ .

To characterize the non-periodic stability associated with walking in rough terrains, we define a progression map between apex states.

**Definition 6** (Stability of Non-Periodic Gaits). We define the stability of non-periodic gaits as the progression map,  $\Phi$ , that takes the robot's center of mass from one desired apex state,  $(\dot{x}_{\text{apex}_q}, z_{\text{apex}_q})$ , to the next one on the phase-space manifold, and via the control input,  $\mathbf{u}$ , i.e.

$$(\dot{x}_{\text{apex}_{q+1}}, z_{\text{apex}_{q+1}}) = \Phi(\dot{x}_{\text{apex}_q}, z_{\text{apex}_q}, \mathbf{u}). \quad (19)$$

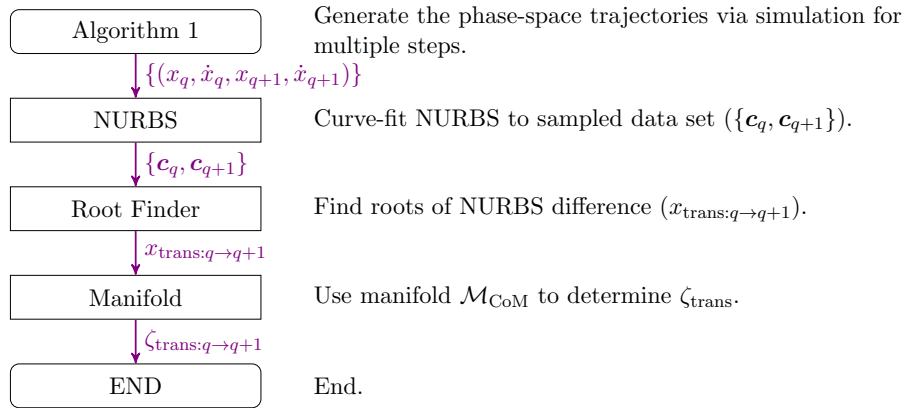


**Fig. 5.** Step transitions. This figure illustrates three types of step transitions in the Sagittal phase-space, associated with  $\sigma$ -isolines: instantaneous step transition in (a), step transition with multi-contact phase in (b) and step transition with invariant bundle in (c). (a) switches between two single contacts instantaneously while (b) has a multi-contact phase. (c) shows several guard alternatives for multi-contact transitions, from the current single-contact manifold value  $\sigma_q$  to the next single-contact step bundle,  $\sigma_{q+2}$ . In particular the invariant bundle bounds,  $\sigma_q = \pm\epsilon$  are shown. The transition bundle in green reattaches to the original manifold,  $\sigma_{q+2} = 0$ , while the transition bundle (in brown) maintains its  $\sigma$  value, i.e.,  $\sigma_{q+2} = \sigma_q$ .

**Definition 7** (Phase Progression Transition Value). A phase progression transition value  $\zeta_{\text{trans}} : \mathcal{Q} \times \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$  is the value of the phase progression variable when the state  $\mathbf{x}_q$  intersects a guard  $\mathcal{G}$ , i.e.,

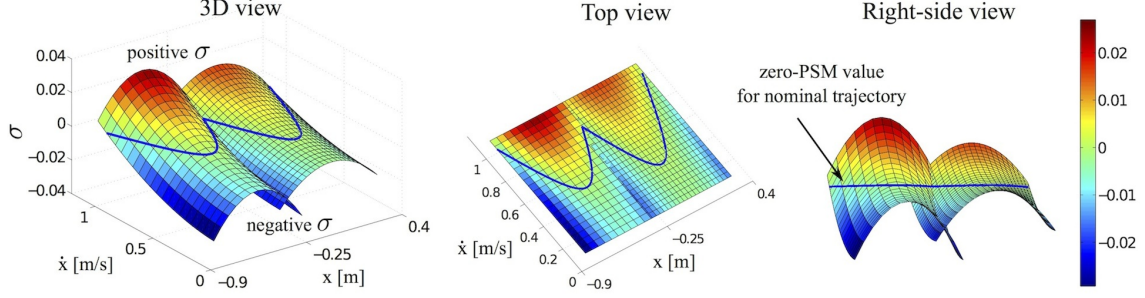
$$\zeta_{\text{trans}} := \inf\{\zeta > 0 \mid \mathbf{x}_q \in \mathcal{G}\}. \quad (20)$$

We propose an algorithm to find transitions between adjacent steps, which occur at  $\zeta_{\text{trans}}$ . We first consider the case of *simple walking steps* as defined in Def. 5. Given known step locations and apex conditions, phase-space curves can be numerically obtained using Algorithm 1. Phase-space curves in the general case have infinite slopes when crossing the zero-velocity axis. Therefore we fit NURBS (non-uniform rational B-splines)<sup>1</sup> to the manifolds of the generated data (see Fig. 6 for an illustration of adjacent step manifolds). Subsequently, finding step transitions just consists on finding the root difference between adjacent NURBS, which reduces to a simple polynomial root-finding problem. The pipeline to find step intersections is shown below.



<sup>1</sup> Different from polynomials, non-rational splines or Bézier curves, NURBS can be used to precisely represent conics and circular arcs by adding weights to control points.





**Fig. 6.** Phase-space manifold isolines. This three-dimensional space demonstrates our phase-space manifold isolines defined in Eq. (22) by the color map. The horizontal plane represents the Sagittal phase-space while the vertical third dimension represents the non-zero  $\sigma$  value in Eq. (22). As we can see, the blue nominal trajectory has a zero  $\sigma$  value. The phase space region above the nominal trajectory has positive  $\sigma$  values while the lower region has negative  $\sigma$  values.

### 5.3 Lateral Foot Placement Algorithm

To complete the 3D walking planner, we formulate a searching strategy for lateral foot placement that complies with the timing of Sagittal step transitions. The main objective of the lateral dynamics is to return the robot's center of mass to a walking center through a semi-periodic cycle. If lateral foot placements are not adequately picked, the lateral behavior will drift away or become unstable. According to Eq. (12), lateral center of mass dynamics are equal to

$$\dot{\mathbf{y}} = \mathcal{F}_{\mathbf{y}}(\mathbf{y}, \mathbf{u}_{\mathbf{y}}) = \begin{pmatrix} \dot{\mathbf{y}} \\ \omega^2(\mathbf{y} - \mathbf{y}_{\text{foot}}) - \frac{\omega^2}{m\mathbf{g}}\boldsymbol{\tau}_{\mathbf{y}} \end{pmatrix}, \quad (21)$$

which can be numerically simulated adapting Algorithm 1 to the lateral dynamics (see Fig. 17 for simulations of lateral dynamics). To generate bounded lateral trajectories, we choose the simple criterion of achieving zero apex

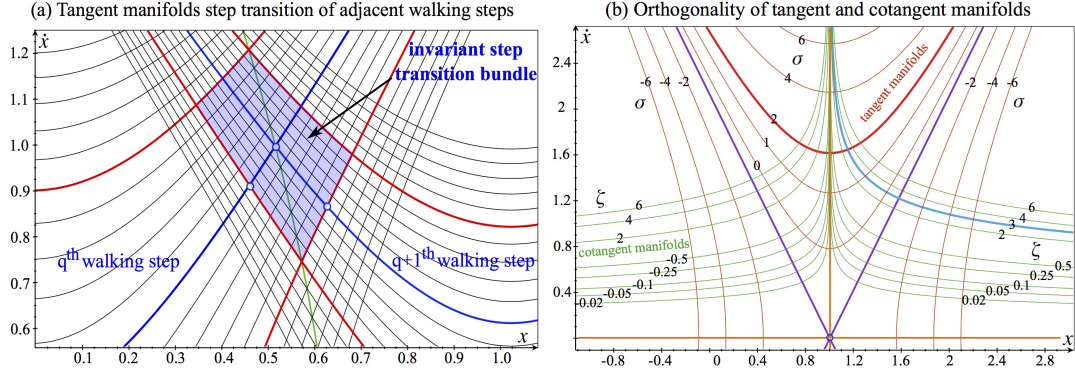
---

#### Algorithm 2 Newton-Raphson Search for Lateral Foot Placement

---

- 1: Initialize iteration index  $n \leftarrow 1$ , maximum iterations  $n_{\text{max}}$ , tolerance  $\dot{y}_{\text{tol}}$  and initial state  $y_{\text{init}}, \hat{y}_{\text{foot}}(1)$
  - 2:  $\dot{y}_{\text{apex}}(1) \leftarrow$  integration of inverted pendulum model given in Eq. (21) with  $\hat{y}_{\text{foot}}(1)$
  - 3: **while**  $n < n_{\text{max}}$  and  $|\dot{y}_{\text{apex}}(n)| > \dot{y}_{\text{tol}}$  **do**
  - 4:    $\hat{y}_{\text{foot}}(n+1) = \hat{y}_{\text{foot}}(n) - \dot{y}_{\text{apex}}(n)/\ddot{y}_{\text{apex}}(n)$  by Newton-Raphson method
  - 5:    $\dot{y}_{\text{apex}}(n+1) \leftarrow$  integration of inverted pendulum in Eq. (21) with  $\hat{y}_{\text{foot}}(n+1)$
  - 6:    $\ddot{y}_{\text{apex}}(n+1) = (\dot{y}_{\text{apex}}(n+1) - \dot{y}_{\text{apex}}(n))/(\hat{y}_{\text{foot}}(n+1) - \hat{y}_{\text{foot}}(n))$
  - 7:    $n \leftarrow n+1$
  - 8: **end while**
- 

lateral velocity,  $\dot{y}_{\text{apex}} = 0$  at the instant when the CoM lateral apex position,  $y_{\text{apex}}$  is located between the two feet. Here  $y_{\text{apex}}$  and  $\dot{y}_{\text{apex}}$  are the lateral center of mass position and velocity when the center of mass crosses the Sagittal apex as defined in Def. 1. Algorithm 2 achieves this objective. In this algorithm,  $\hat{y}_{\text{foot}}(n)$  represents the estimated lateral foot placements in the  $n^{\text{th}}$  search iteration. A foot placement range constraint  $\hat{y}_{\text{foot},\text{min}} \leq \hat{y}_{\text{foot}} \leq \hat{y}_{\text{foot},\text{max}}$  and the maximum iteration step constraint  $n < n_{\text{max}}$  are also provided. Examples of usage are given in Fig. 17.



**Fig. 7.** The left subfigure shows the tangent manifolds for two simple walking steps. Nominal phase space manifolds are shown in blue color. Their intersection corresponds to  $\zeta_{\text{trans}}$  and the guard,  $\mathcal{G}_{q \rightarrow q+1}$ . Shown in red are manifolds of  $\sigma = \pm\epsilon$ . For the current  $q^{\text{th}}$  step, we can use the  $-\epsilon$  manifold of the next bundle as the guard  $\mathcal{G}_{q \rightarrow q+1} = \{(x, \dot{x}) \mid \sigma_{q+1} = -\epsilon\}$ . The subfigure on the right shows the orthogonal tangent and cotangent manifolds (*isoparametric curves*) in phase-space. The tangent manifold (red lines) are shown as lines of constant  $\sigma$  as defined in Eq. (23). Thick lines in purple are the asymptotes of tangent manifolds and the thick red line represents the specific manifold with  $\sigma = 2$ . The asymptotes intercept the  $(x_{\text{foot}}, 0)$  saddle point, where  $x_{\text{foot}}$  is the Sagittal foot position. The point noted with a circle is the origin of the  $(\zeta, \sigma)$  (transformed) space. The cotangent manifold (green lines) are lines of constant  $\zeta$  that are orthogonal at every point to the constant  $\sigma$ -manifolds. Thick lines in orange are the asymptotes of cotangent manifolds and the thick cyan line represents a specific manifold with  $\zeta = 3$ . The vertical asymptote represents the manifold with  $\zeta = 0$ .

## 6 Phase-Space Manifold Analysis

In this section, we focus on formulating a metric to measure deviations from the phase-space manifolds of planned trajectories derived in the previous section. A phase-space sensitivity norm is also formulated to study the effect of disturbances. Various disturbance patterns and suggested recovery strategies are considered.

**Proposition 1** (Phase-Space Tangent Manifold). *Given the prismatic inverted pendulum dynamics of Eq. (15) with an initial condition  $(x_0, \dot{x}_0)$  and foot placement  $x_{\text{foot}}$ , the phase-space tangent manifold is*

$$\sigma = (x_0 - x_{\text{foot}})^2 \left( 2\dot{x}_0^2 - \dot{x}^2 + \omega^2(x - x_0)(x + x_0 - 2x_{\text{foot}}) \right) - \dot{x}_0^2(x - x_{\text{foot}})^2 + \dot{x}_0^2(\dot{x}^2 - \dot{x}_0^2)/\omega^2, \quad (22)$$

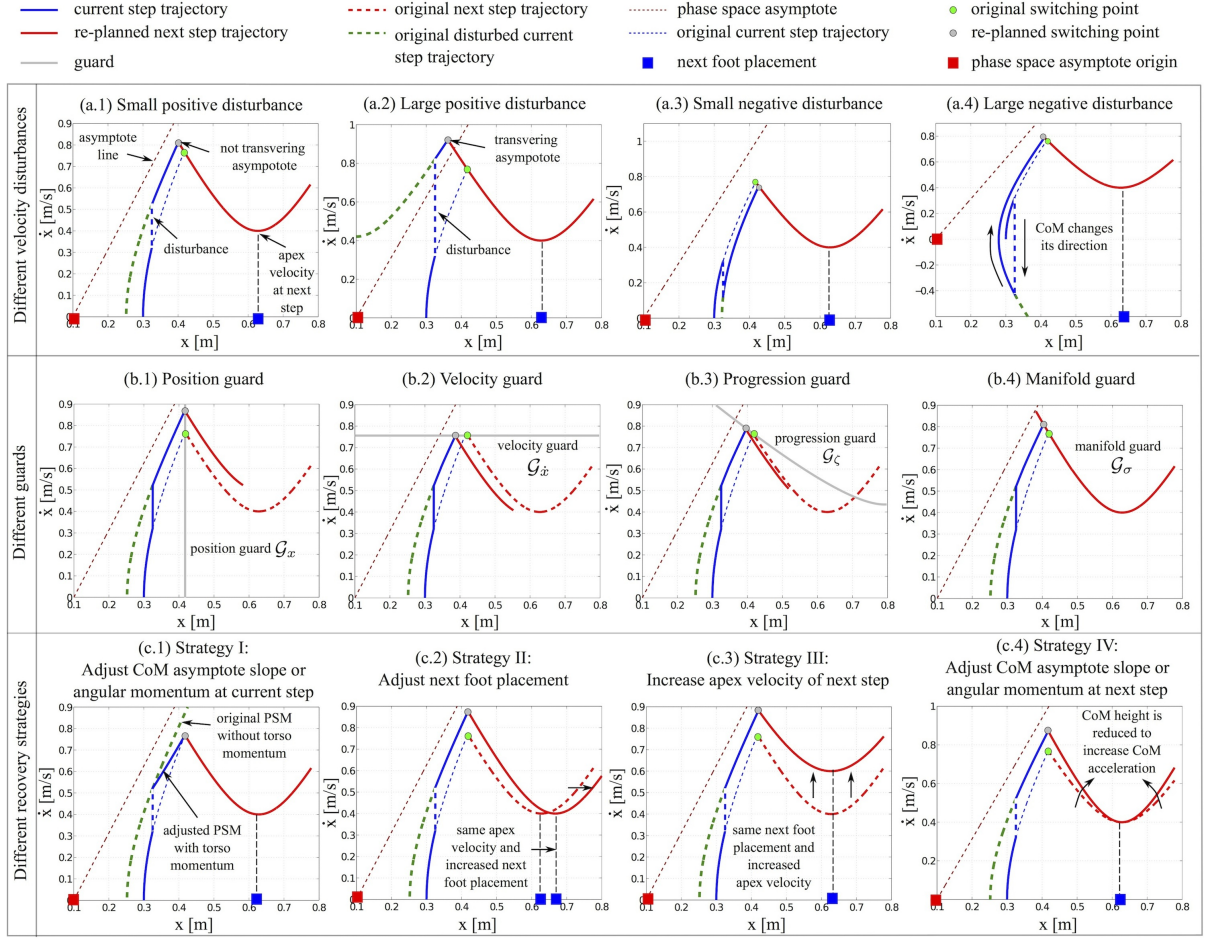
where the condition  $\sigma = 0$  is equivalent to the nominal phase-space manifold. Furthermore,  $\sigma$  represents the Riemannian distance to the estimated locomotion phase-space trajectories.

*Proof.* Given in the Appendix E. □

If we use the apex conditions as initial values, i.e.  $(x_0, \dot{x}_0) = (x_{\text{foot}}, \dot{x}_{\text{apex}})$ , the tangent manifold becomes

$$\sigma(x, \dot{x}, \dot{x}_{\text{apex}}, x_{\text{foot}}) = \frac{\dot{x}_{\text{apex}}^2}{\omega^2} \left( \dot{x}^2 - \dot{x}_{\text{apex}}^2 - \omega^2(x - x_{\text{foot}})^2 \right). \quad (23)$$

Since the tangent manifold is considered as a trajectory deviation metric in the phase-space, we will use it in the next section as a feedback control parameter to ensure robustness. Fig. 6 provides an illustration of the value of  $\sigma$  as a function of the state. The same type of analysis can be done for lateral trajectory deviations using the lateral pendulum dynamics of Eq. (21).



**Fig. 8.** Disturbance pattern, guard and recovery strategy classification. Four different velocity disturbance cases are shown in subfigures (a.1)–(a.4). The second row shows four proposed guards for the next step transition while the last row provides four recovery strategies.

**Proposition 2** (Phase-Space Cotangent Manifold). *Given the pendulum system of Eq. (15), the cotangent manifold is equal to*

$$\zeta = \zeta_0 \left( \frac{\dot{x}}{\dot{x}_0} \right)^{\omega^2} \frac{x - x_{\text{foot}}}{x_0 - x_{\text{foot}}}, \quad (24)$$

*and represents the arc length along the tangent manifold of Eq. (23), with an initial value  $\zeta_0$ .*

*Proof.* Given in the Appendix F. □

Illustration of the tangent manifold and cotangent manifold are given in Fig. 7. In robust control theory [Zhou et al. (1996)], input-output system response can be evaluated through the use of system norms. In this spirit, we define a new norm that characterizes sensitivity to disturbances of our non-periodic gaits, as

**Definition 8** (Phase-Space Sensitivity Norm). *Given a disturbance  $d$ , the phase-space sensitivity norm is defined as*

$$\kappa\left(\sigma(x_{\zeta_d}, \dot{x}_{\zeta_d})\right) = \left(\frac{1}{\zeta_{\text{trans}} - \zeta_d} \int_{\zeta_d}^{\zeta_{\text{trans}}} \sigma(x_{\zeta}, \dot{x}_{\zeta})^2 d\zeta\right)^{1/2}, \quad (25)$$

where,  $\zeta_d$  corresponds to the phase value when a disturbance occurs and  $\zeta_{\text{trans}}$  is the phase transition defined in Def. 7 for a given step.

Contrary to other sensitivity norms [Hobbelen and Wisse (2007); Hamed et al. (2015)], our gait norm evaluates disturbance sensitivity for non-periodic gaits. It does so by explicitly accounting for disturbance magnitude and for the instant where disturbances occur. And it does not rely on approximate linearization nor Taylor series expansion as the previous periodic gait norms require. We will use the proposed norm in the control section for dynamic programming.

Let us consider various types of disturbances and corresponding recovery strategies. Disturbances can be categorized based on four characteristics: (1) the disturbance direction, (2) the disturbance magnitude, (3) the terminal asymptote-region, and (4) the change of the motion direction. Fig. 8 (a.1)-(a.4) illustrates those four scenarios, respectively. Here, we assume that the disturbances are impulses that change the velocity instantly<sup>2</sup>. (a.2) has a larger positive disturbance than (a.1) such that velocity after the disturbed trajectory crosses the asymptote of the inverted pendulum model. On the other hand, (a.3) has a smaller negative disturbance such that velocity after disturbance keeps the same direction while (a.4) does not. More disturbance scenarios could be defined, depending on specific phase-space state locations and disturbance characteristics. Given these disturbed phase-space trajectories, new step transition strategies need to be considered. Here we propose four types of guard strategies for next step transition in Fig. 8 (b.1)-(b.4). The guards shown are: position guard  $\mathcal{G}_x$  (vertical line), velocity guard  $\mathcal{G}_{\dot{x}}$  (horizontal line), progression guard  $\mathcal{G}_{\zeta}$  ( $\zeta$ -isoline), and manifold guard  $\mathcal{G}_{\sigma}$  ( $\sigma$ -isoline). We find each guard such that they have the same transition point for the nominal phase space manifold (PSM). Although this guard recovering strategy forces the motion to be adjusted, it might not be sufficient. If that is the case, we consider designing more recovery strategies by properly using control inputs. In the last four subfigures of Fig. 8, four recovery strategies are illustrated. These strategies are inspired by observations of human walking behaviors [Hofmann (2006); Kuo and Zajac (1992); Abdallah and Goswami (2005)] and by the experiences gained during extensive simulations. In order to fulfill those recovery strategies, a proper control policy will need to be designed.

## 7 Hybrid Control Strategy under Disturbances

This section formulates a two-stage control procedure to recover from disturbances. When a disturbance occurs, the robot's CoM deviates from the planned phase-space manifolds obtained via Algorithm 1. We use dynamic programming to find an optimal policy of the continuous control variables for recovery, and, when necessary, we re-plan feet placements from their initial locations based on the guards defined in Fig. 8. Our proposed controller, relies on the distance metric of Eq. (23) to steer the robot current's trajectory to the planned manifolds.

---

<sup>2</sup> The disturbance could also be impulses changing the position or a combination of both position and velocity. In any case, the proposed disturbance characteristics and recovery strategies are still valid.

### 7.1 Dynamic Programming-Based Optimal Control

This subsection shows a proposed dynamic programming-based controller for the continuous control of the Sagittal locomotion dynamics. A similar controller can be formulated for the lateral CoM behavior. To robustly track the planned CoM manifolds, we minimize a finite phase quadratic cost function and solve for the continuous control parameters, i.e.

$$\begin{aligned} \min_{\mathbf{u}_x^c} \quad & \mathcal{V}_N(q, \mathbf{x}_N) + \sum_{n=0}^{N-1} \eta^n \mathcal{L}_n(q, \mathbf{x}_n, \mathbf{u}_x^c) \\ \text{subject to : } \quad & \dot{\mathbf{x}} = \mathcal{F}_x(\mathbf{x}, \mathbf{u}_x^c, d), \\ & \omega^{\min} \leq \omega \leq \omega^{\max}, \\ & \tau_y^{\min} \leq \tau_y \leq \tau_y^{\max}, \end{aligned} \quad (26)$$

where  $\mathbf{u}_x^c = \{\omega, \tau_y\}$  corresponds to the continuous variables of the hybrid control input  $\mathbf{u}_x$  of Eq. (15),  $0 \leq \eta \leq 1$  is a discount factor,  $N$  is the number of discretized stages until the next step transition,  $\zeta_{\text{trans}}$ , the terminal cost is,  $\mathcal{V}_N(q, \mathbf{x}_N) = \alpha(\dot{x}(\zeta_{\text{trans}}) - \dot{x}(\zeta_{\text{trans}})^{\text{nom}})^2$ . Here,  $\dot{x}(\zeta_{\text{trans}})$  is the disturbed terminal velocity which happens to be chosen at the instant of the next step transition, and  $\dot{x}(\zeta_{\text{trans}})^{\text{nom}}$  is the nominal transition velocity. The first equality constraint  $\mathcal{F}_x(\cdot)$  is defined by the PIPM dynamics of Eq. (15) with an extra input disturbance  $d$ . Additionally,  $\mathcal{L}_n$  is the one step cost-to-go function at the  $n^{\text{th}}$  stage defined as

$$\mathcal{L}_n(q, \mathbf{x}_n, \mathbf{u}_x^c) = \int_{\zeta_n}^{\zeta_{n+1}} [\beta\sigma^2 + \Gamma_1\tau_y^2 + \Gamma_2(\omega - \omega^{\text{ref}})^2] d\zeta, \quad (27)$$

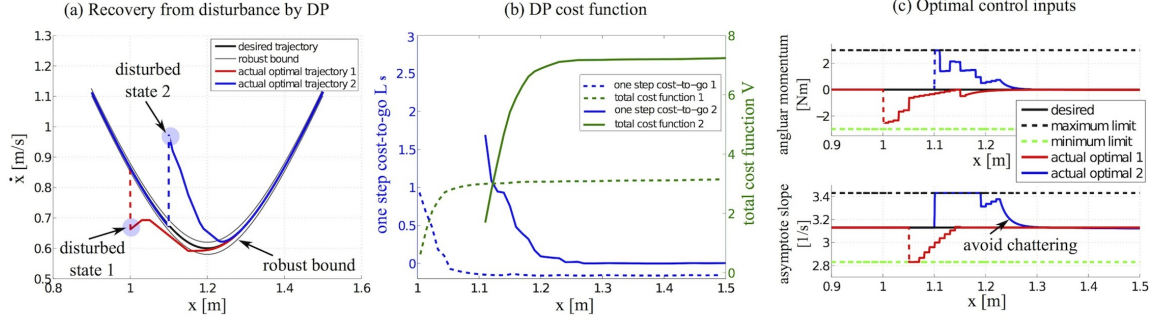
where,  $\sigma$  is the tangent manifold of Eq. (22) used as a feedback control parameter,  $\zeta_n$  and  $\zeta_{n+1}$  are the starting and ending phase progressions for the  $n^{\text{th}}$  stage,  $\alpha$ ,  $\beta$ ,  $\Gamma_1$  and  $\Gamma_2$  are weights, and  $\omega^{\text{ref}}$  is the reference phase space asymptote slope given in Algorithm 1. To avoid chattering effects in the neighborhood of the planned manifold, a boundary layer is defined and used to saturate the controls, i.e.

$$\mathbf{u}_x^{c'} = \begin{cases} \mathbf{u}_x^c & |\sigma| > \epsilon \\ \frac{|\sigma|}{\epsilon} \mathbf{u}_x^{c,\epsilon} + \frac{\epsilon - |\sigma|}{\epsilon} \mathbf{u}_x^{c,\text{ref}} & |\sigma| \leq \epsilon \end{cases} \quad (28a)$$

$$|\sigma| \leq \epsilon \quad (28b)$$

where  $\epsilon$  corresponds to the boundary value of an invariant bundle  $\mathcal{B}(\epsilon)$  as defined in Def. 2,  $\mathbf{u}_x^{c,\epsilon} = \{\omega^\epsilon, \tau_y^\epsilon\}$  are control inputs at the instant when the trajectory enters the invariant bundle  $\mathcal{B}(\epsilon)$ ,  $\mathbf{u}_x^{c,\text{ref}}$  are nominal control inputs defined in Algorithm 1. A proof of smoothness of the above control saturation function is discussed in [Utkin (2013)]. As Eq. (28) shows, when  $|\sigma| \leq \epsilon$ , the control effort,  $\mathbf{u}_x^{c'}$  is scaled between  $\mathbf{u}_x^{c,\epsilon}$  and  $\mathbf{u}_x^{c,\text{ref}}$ . This control law is composed of an “inner” and an “outer” controller. The “outer” controller steers states into  $\mathcal{B}(\epsilon)$  while the “inner” controller maintains states within  $\mathcal{B}(\epsilon)$ . Recovery trajectories are shown in Fig. 9 for two scenarios in the presence of random disturbances.

Since the control inputs are constrained within a desired range, i.e.  $\mathbf{u}_x^c \in \mathbf{u}_x^{c,\text{range}}$ , we re-define the finite-phase control-dependent recoverability bundle. Given an acceptable deviation  $\epsilon_0$  from the manifold, the practical invariant bundle is  $\mathcal{B}(\epsilon_0)$ . The control policy in Eq. (28) generates a control-dependent practical recoverability bundle (a.k.a.,



**Fig. 9.** Chattering-free recoveries from disturbance by the proposed optimal recovery continuous control law. Subfigure (a) shows two random disturbances, where disturbed state 1 has a positive impulse while the disturbed state 2 has a negative impulse. Control variables are piece-wise constant within one stage as shown in subfigure (c). In these simulations, angular momentum control range is  $[-3, 3]$  Nm and phase space asymptote slope range is  $[2.83, 3.43]$  1/s. Other parameters are shown in Table 2.

Table 2: Dynamic Programming Parameters

Parameter	Range	Parameter	Range
desired momentum $\tau_y^{\text{ref}}$	0 Nm	desired asymptote slope $\omega^{\text{ref}}$	3.13 1/s
angular momentum range $\tau_y^{\text{range}}$	$[-3, 3]$ Nm	asymptote slope range $\omega^{\text{range}}$	$[2.83, 3.43]$ 1/s
foot placement $x_{\text{foot}}$	1.2 m	mass $m$	1 kg
stage range	$[0.9, 1.5]$ m	state range	$[0.03, 1.5]$ m/s
stage resolution $\text{stage}_{\text{res}}$	0.01 m	state resolution $\text{state}_{\text{res}}$	0.01 m/s
disturbed initial state $s_{\text{initial}}$	(1.1 m, 0.7 m/s)	desired apex velocity $\dot{x}_{\text{apex}}$	0.6 m/s
weighting scalar $\Gamma_1$	5	weighting scalar $\Gamma_2$	5
weighting scalar $\beta$	$4 \times 10^4$	weighting scalar $\alpha$	100

region of attraction to the “boundary-layer”) defined as

$$\mathcal{R}(\epsilon, \zeta_{\text{trans}}) = \left\{ \mathbf{x}_\zeta \in \mathbb{R}^2, \quad \zeta_0 \leq \zeta \leq \zeta_{\text{trans}} \mid \mathbf{x}_{\zeta_{\text{trans}}} \in \mathcal{B}(\epsilon), \quad \mathbf{u}_\mathbf{x}^c \in \mathbf{u}_\mathbf{x}^{c, \text{range}} \right\}. \quad (29)$$

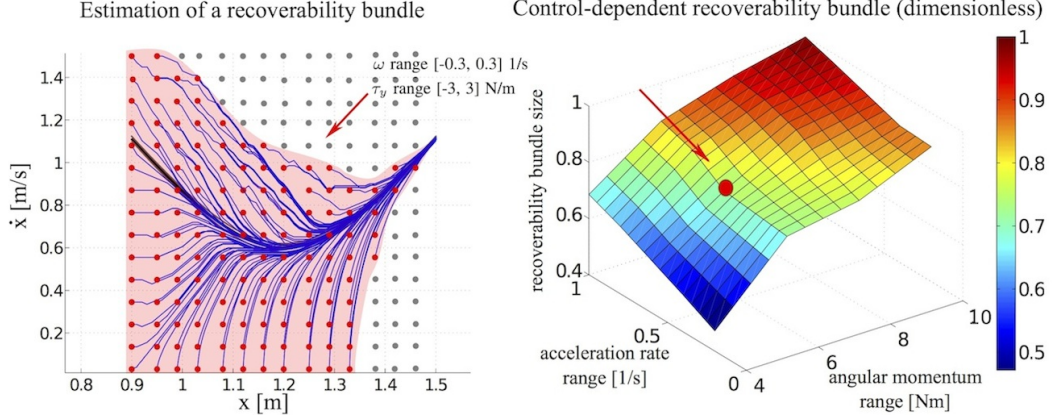
**Theorem** (Existence of Recoverability Bundle). *Given a Lyapunov function  $V = \sigma^2/2$ , a phase progression transition value  $\zeta_{\text{trans}}$ , and the control policy in Eq. (28), a recoverability bundle  $\mathcal{R}(\epsilon, \zeta_{\text{trans}})$  exists and can be bounded by a maximum tube radius  $\sigma_0^{\text{max}}$ .*

*Proof.* We will use a Lyapunov function to prove the existence of  $\mathcal{R}(\epsilon, \zeta_{\text{trans}})$ . First, let us define  $V = \sigma^2/2$  based on Eq. (28a). If there exists a control policy such that  $\exists \sigma_0 > \epsilon, \sigma_{\text{trans}} \leq \epsilon$ , then,  $\mathcal{R}(\epsilon, \zeta_{\text{trans}})$  is composed of the range of values  $(x, \dot{x})_\zeta$ ,  $\zeta_0 \leq \zeta \leq \zeta_{\text{trans}}$ , such that  $V_{\text{trans}} = \sigma_{\text{trans}}^2/2 \leq \epsilon^2/2$ . Taking the derivative of  $V$  along the pendulum dynamics in Eq. (12), we have

$$\begin{aligned} \dot{V} &= \sigma \dot{\sigma} = \sigma \dot{x}_{\text{apex}}^2 \left( -2\dot{x}(x - x_{\text{foot}}) + 2\dot{x}\ddot{x}/\omega^2 \right) = \sigma \dot{x}_{\text{apex}}^2 \left( -2\dot{x}(x - x_{\text{foot}}) + 2\dot{x}((x - x_{\text{foot}}) - \frac{\tau_y}{mg}) \right) \\ &= -\frac{2\dot{x}_{\text{apex}}^2 \sigma \dot{x} \tau_y}{mg} = -\frac{2\sqrt{2}\dot{x}_{\text{apex}}^2 \dot{x} \tau_y \cdot \text{sign}(\sigma)}{mg} \sqrt{V} \leq 0. \end{aligned} \quad (30)$$

which proves the stability (i.e., attractiveness) of  $\sigma = 0$ . For example, consider the case of walking forward,  $\dot{x} > 0$ . Then, as long as  $\sigma \cdot \tau_y > 0$ , i.e., the pitch torque has the same sign as  $\sigma$ , attractiveness is guaranteed. That is, if  $\sigma > 0$  (the robot moves forward faster than expected), then we need  $\tau_y > 0$  to slow down, and vice-versa. To estimate





**Fig. 10.** Estimation of dimensionless control-dependent recoverable region. In the left figure, disturbed states are sampled in a discretized grid and the shaded region represents the recoverability bundle. As it is shown, a larger recoverable region is achieved in the beginning of the phase (i.e., before the apex state). In the ending phase, the recoverable region shrinks to the invariant bundle. Here the control constraint is:  $\omega \in [-0.3, 0.3]$  1/s and  $\tau_y \in [-3, 3]$  N/m. The right figure shows the dependence of the size of the recoverable region with respect to the allowed control ranges. For visualization, the control range in the horizontal axes are labeled as values  $r$ , which implies the range is  $[-r/2, r/2]$ .

$\mathcal{R}(\epsilon, \zeta_{\text{trans}})$ , we can use the optimal control policy proposed in Eq. (28), defining an “optimal” recoverability bundle; or (ii) the maximum control inputs (without any regards to optimality) obtained by selecting the bounds  $\mathbf{u}_x^{c, \text{range}}$ , defining the “maximum” recoverability bundle:

*Case I: DP based Control.* If  $\tau_y$  is solved by the optimization of Eq. (26), we get  $|\tau_y| > |\tau_y^\epsilon|$ . Then Eq. (30) becomes

$$\dot{V} < -\frac{2\sqrt{2}\dot{x}_{\text{apex}}^2 \dot{x} |\tau_y^\epsilon|}{mg} \sqrt{V} < 0. \quad (31)$$

*Case II: Supremum (Bang-Bang) Control.* If we design  $\tau_y = \tau_y^{\max} \text{sign}(\dot{x})$  and  $\dot{x} > 0$ , then,

$$\dot{V} = -\frac{2\sqrt{2}\dot{x}_{\text{apex}}^2 \dot{x} \tau_y^{\max} \text{sign}(\dot{x})}{mg} \sqrt{V} = -\frac{2\sqrt{2}\dot{x}_{\text{apex}}^2 \dot{x} \tau_y^{\max}}{mg} \sqrt{V} < 0. \quad (32)$$

Note that, the bounded  $\dot{V}$  in Eqs. (31) and (32) have similar forms and can be combined into a common integral equation of form

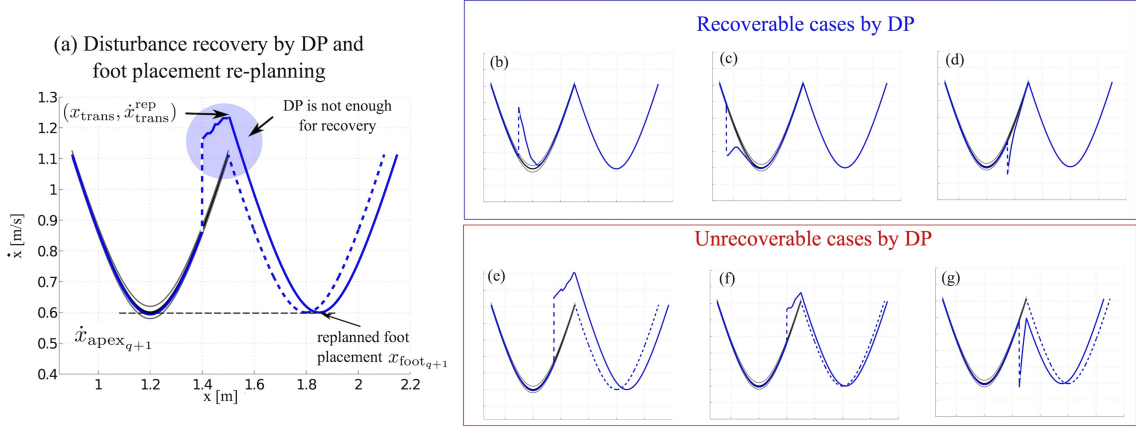
$$\int_{V_0}^{V_{\text{trans}}} \frac{dV}{\sqrt{V}} = -\int_{t_0}^{t_{\text{trans}}} \mu \dot{x} \tau_y dt = -\mu \tau_y (x_{\text{trans}} - x_0), \quad (33)$$

where  $\mu = (2\sqrt{2}\dot{x}_{\text{apex}}^2)/(mg)$ ,  $\tau_y = \tau_y^\epsilon$  for Case I while  $\tau_y = \tau_y^{\max}$  for Case II. Eq. (33) can be solved using common integral rules to yield

$$\sqrt{V_0} = \sqrt{V_{\text{trans}}} + \frac{1}{2}\mu \cdot (x_{\text{trans}} - x_0) \cdot \tau_y. \quad (34)$$

Since  $V_0 = \sigma_0^2/2$ ,  $V_{\text{trans}} = \sigma_{\text{trans}}^2/2 \leq \epsilon^2/2$ , we get

$$\sigma_0 \leq \epsilon + \frac{\sqrt{2}}{2}\mu \cdot (x_{\text{trans}} - x_0) \cdot \tau_y = \sigma_0^{\max}, \quad (35)$$



**Fig. 11.** Recovery from a disturbance by re-planning Sagittal foot placement. In this case, the next apex velocity is given *a priori* and maintained despite the disturbance. In subfigures (b)-(d), first-stage continuous DP control is sufficient to achieve the recovery while in the cases of subfigures (e)-(g) it is not. The latter cases occur when either disturbance occurs too close to the transition or is too large. In these cases, a new next foot placement is automatically re-planned based on Eq. (37).

where  $\sigma_0^{\max}$  defines the maximum tube radius. Therefore we can re-write the recoverability bundle of Eq. (29) using this new tube radius as:

$$\mathcal{R}(\epsilon, \zeta_{\text{trans}}) = \left\{ \mathbf{x}_\zeta \in \mathbb{R}^2, \zeta_0 \leq \zeta \leq \zeta_{\text{trans}} \mid \epsilon \leq \sigma_0^{\max} \right\}. \quad (36)$$

The existence of recoverability bundle is proved and a maximum tube radius is provided.  $\square$

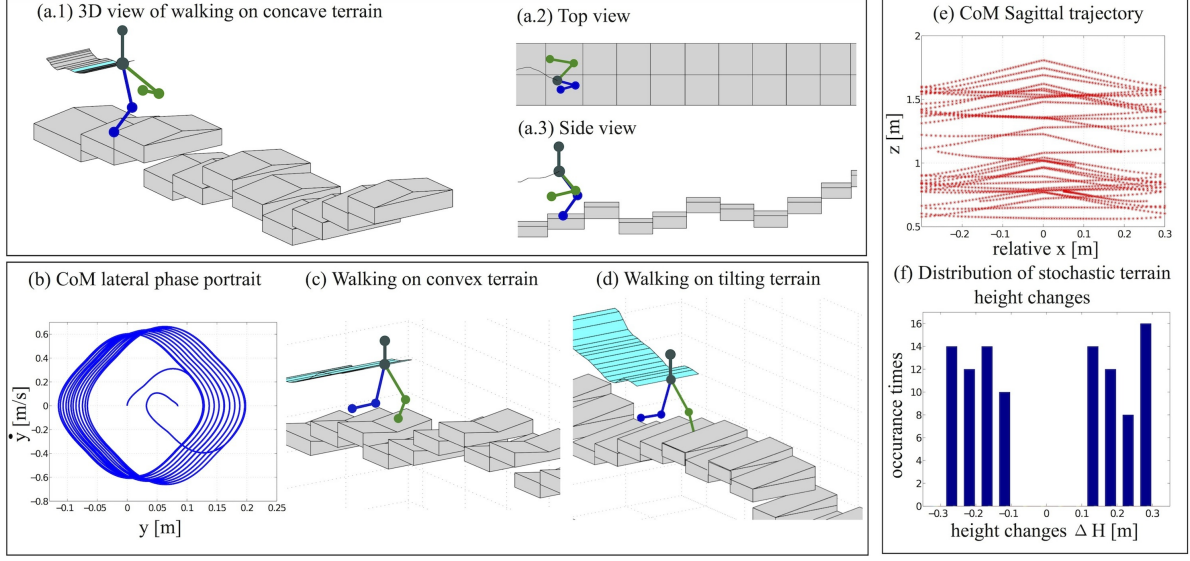
Since Eq. (31) has an inequality bound while Eq. (32) has an equality bound, DP based control is an optimal but conservative estimation of the true recoverability bundle while supremum control is an accurate but non-optimal estimation for the recoverability bundle. Our study aims at optimal performance, and therefore the control policy generated from dynamic programming will be used to estimate the recoverability bundle. To estimate  $\mathcal{R}(\epsilon, \zeta_{\text{trans}})$ , we perform a grid sampling from the initial condition  $\mathbf{x}_{\zeta_0}$ , based on the ranges of Table 2. Then we execute the optimization of Eq. (26) for each sampled  $\mathbf{x}_{\zeta_0}$  (treated as a realization) and repeat this procedure for all  $\mathbf{x}_{\zeta_0}$  in the grid. The feasible realizations of recovery trajectories (i.e. the convergence into  $\mathcal{B}(\epsilon)$  before  $\zeta_{\text{trans}}$ ) constitute the recoverability bundle<sup>3</sup>. An example of an estimated recoverability bundle is shown in Fig. 10 (a).

## 7.2 Discrete Foot Placement Control

When the disturbance is large enough to move the state outside of the recoverability manifold, the controller will not be able to recover to the invariant bundle. We propose to use the guard strategies discussed in Section 6 for recovery. As a case study, let us use the position guard strategy and re-plan the foot placement for the next step as was illustrated in Fig. 8 (c.2). In that strategy, the next apex velocity,  $\dot{x}_{\text{apex}_{q+1}}$ , is kept as planned. Hence, we analytically solve for a new foot placement based on the PSM given in Eq. (23) such that the apex velocity is achieved. Let us define the re-planned phase-space transition state as  $(x_{\text{trans}}, \dot{x}_{\text{trans}}^{\text{rep}})$ , where only velocity  $\dot{x}_{\text{trans}}^{\text{rep}}$

<sup>3</sup> Here, only positive disturbed state  $\dot{x} > 0$  is considered. Recovery of  $\dot{x} < 0$  could be achieved in a similar manner. All that is needed is to integrate phase-space trajectories in a backward pattern, detect the PSM deviation by Eq. (22) and look up an offline DP policy table designed for backward walking.





**Fig. 12.** Traversing various rough terrains. The subfigures on the left block show dynamic locomotion over rough terrains with varying heights. The block on the right shows the height variation distribution over 100-steps.

varies. Since  $\dot{x}_{\text{apex}_{q+1}}$  is unchanged, the adjusted Sagittal foot placement  $x_{\text{foot}_{q+1}}$  is solved by simple manipulation of Eq. (23)

$$x_{\text{foot}_{q+1}} = x_{\text{trans}} + \frac{1}{\omega} (\dot{x}_{\text{trans}}^{\text{rep}2} - \dot{x}_{\text{apex}_{q+1}}^2)^{1/2}. \quad (37)$$

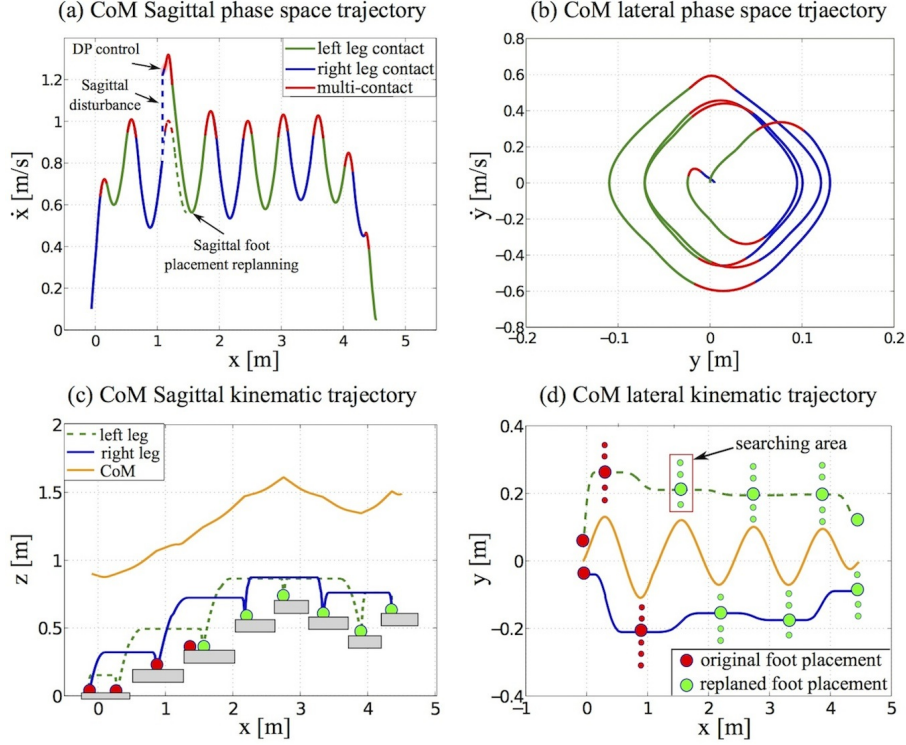
For forward walking,  $x_{\text{foot}_{q+1}} > x_{\text{trans}}$ , prompting us to ignore the solution with the negative square root. To evaluate the performance of this step re-planning method, we consider the six disturbances scenarios of Fig. 11. The top three scenarios are recoverable using the DP-based continuous controller that we presented earlier. In the bottom three scenarios the disturbance occurs too close to the transition or is too large and therefore requires the foot placement re-planner described above to be executed. Once foot placements have been re-planned in the Sagittal direction, lateral foot placements are re-planned using Algorithm 2.

To conclude, the two-stage procedure discussed in this section constitutes the core process of our robust-optimal phase-space planning strategy. The combine locomotion planning procedure is shown in Algorithm 3 in the Appendix. The computational burden of our technique is minimal because: (I) Once a disturbance is detected, the offline DP policy table is quickly looked up rather than re-computed. (II) If disturbances are too high, Eq. (37) will quickly yield a new proper foot location, relying on the efficiency of a closed analytical solution.

## 8 Simulated Results

### 8.1 Dynamic Walking over Rough Terrains

We validate the versatility of our phase-space planning and control strategy by performing locomotion over terrains with height variations randomly generated. Three challenging terrains are tested as shown in Fig. 12: (a) a terrain with concave steps, (c) a terrain with convex steps and (d) a terrain with inclined steps. The height variation,  $\Delta h_k$ ,



**Fig. 13.** Recovery from Sagittal plane disturbance. To recover from a Sagittal push, the planner uses both DP continuous control and discrete foot placement re-planning in a sequential manner. ● denotes the pre-defined foot placement before the disturbance while ● denotes the re-planned foot placement after the disturbance.

of two consecutive steps is randomly generated based on the uniform distribution,

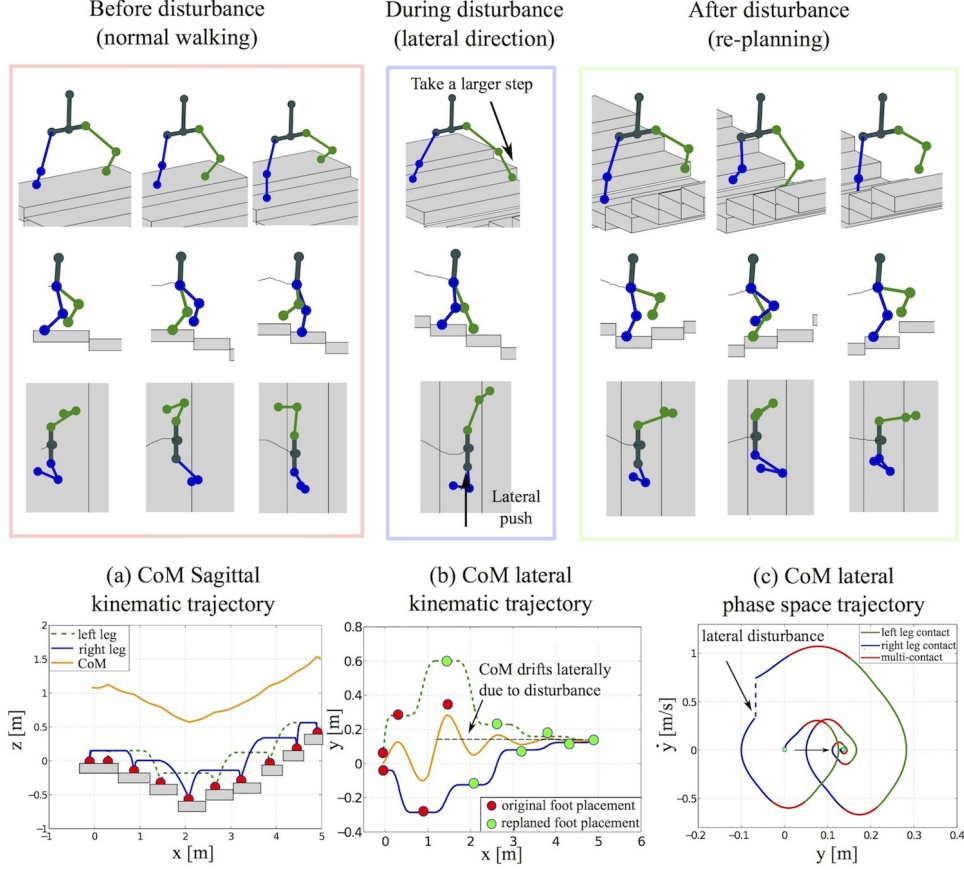
$$\Delta h_k = h_{k+1} - h_k \sim \mathcal{U}\{(-\Delta h_{\max}, -\Delta h_{\min}) \cup (\Delta h_{\min}, \Delta h_{\max})\}, \quad (38)$$

where,  $h_k$  represents the height of the  $k^{\text{th}}$  step,  $\Delta h_{\min} = 0.1$  m,  $\Delta h_{\max} = 0.3$  m. A  $10^\circ$  tilt angle is used for the slope of the steps. Foot placements are chosen a priori using simple kinematic rules and considering the length of the terrain steps. Separately, we choose apex velocities to be around 0.6 m/s, plus minus a small increment depending on the height change. Finally we choose simple CoM path manifolds that conform to the steps. We then apply the planning pipeline outlined in previous sections including, the generation of trajectories based on Algorithm 1 and the search for step transitions based on the procedures of Section 5.2.

Fig. 12 (a) shows a snapshot of bipedal walking on the terrain with concave steps. The lateral CoM phase portrait in Fig. 12 (b) shows stable walking over 25 steps. Fig. 12 (c) and (d) show other types of rough terrains also tested in simulation over 100 steps. Fig. 12 (e) shows the CoM path manifolds we choose to step over the terrain. The bar graph in Fig. 12 (f) shows the distribution of the height of the randomly generated terrain.

## 8.2 Dynamic Walking under External Disturbances

*Recovery from Disturbance on the Sagittal Plane* We first make the robot walk on a terrain based on the planning algorithms described in the paper. We then apply a pushing force in the Sagittal direction, which causes an instantaneous velocity jump as shown in Fig. 13 (a). This disturbance is quite large such that the robot's state cannot



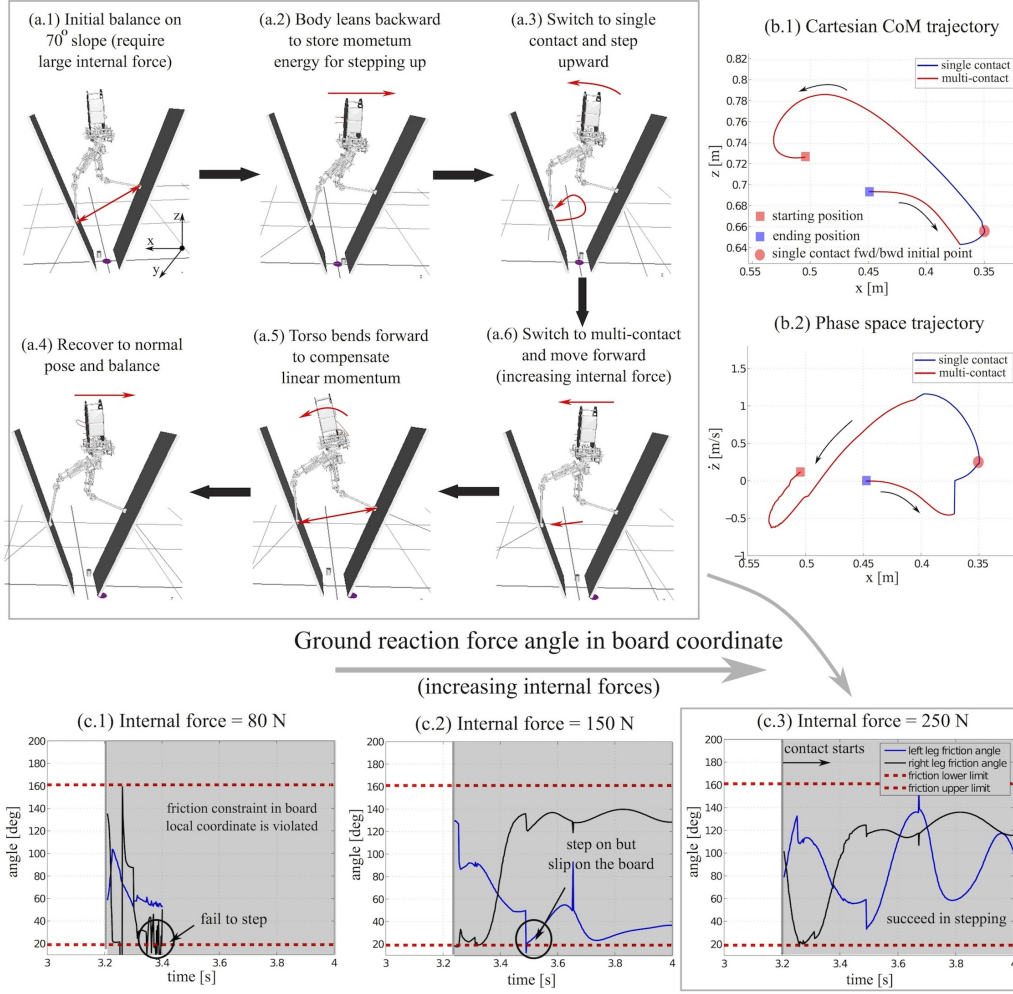
**Fig. 14.** Recovery from lateral plane disturbance. When the lateral disturbance occurs, the foot placement is re-planned to avoid falling down. Correspondingly, the CoM trajectory drifts to the left side as shown in (b) while a CoM velocity jump appears in (c). After the disturbance, the CoM trajectory is re-generated based on this new lateral foot placement to achieve stable walking.

recover to its nominal PSM using the proposed optimal controller. Thus, the foot location re-planning strategy previously described is executed. The dashed line in Fig. 13 (a) represents the original phase-space trajectory while the solid line represents the re-planned trajectory. Also, instead of an instantaneous step transition, a multi-contact transition is used as described in the Appendix D.

*Recovery from Disturbance on the Lateral Plane* For this simulation, we make the robot once more walk on the rough terrain according to its nominal plan. Then, in its third step, we apply a lateral disturbance as shown in Fig. 14 (b) and (c). To deal with this disturbance, a new lateral foot placement is re-planned according to Algorithm 2.

### 8.3 Bouncing over A Disjointed Terrain

A more challenging locomotion scenario is explored using a disjointed terrain. The slope of the surfaces is  $70^\circ$ . The goal is to step up over the surfaces by bouncing over the terrain. A physics based dynamic simulation called SrLib is used for validation and a whole body operational space controller [Sentis et al. (2010)] is implemented to follow the locomotion plans.



**Fig. 15.** Bouncing over a disjointed terrain. In this simulation, a biped balances on a steep disjointed terrain and dynamically bounces upwards. Internal force and angular momentum are controlled properly to achieve this motion. Subfigure (b) shows the 2D Cartesian CoM trajectory and the  $x - \dot{z}$  phase portrait. (c) shows contact reactions for three different desired internal forces.

Snapshots of a one-step bouncing behavior are shown in Fig. 15 (a). To successfully bounce over the terrain, we use a CoM path manifold, shown in Fig. 15 (b.1), that mimics that of a pre-recorded human [Sentis and Slovic (2011)]. During the multi-contact phase, we apply a 250 N internal tension force, shown in Fig. 15 (c), between the two surfaces to avoid sliding down due to the weight of the robot. Our planner operates in the  $x - \dot{z}$  state as shown in Fig. 15 (b.2). This is more convenient as the  $\dot{z}$  reveals the moment at which the center of mass starts falling down. More details on this strategy are discussed in [Sentis and Slovic (2011); Zhao et al. (2015)]. Note that an apex state is not used in the same way that we defined for previous rough terrain locomotion tests. Instead, we define a priori a desired keyframe state, shown as a red circle in Fig. 15 (b.2).

## 9 Discussions and Conclusions

The main focus of this paper has been on addressing the needs for planning non-periodic bipedal locomotion behaviors. These types of behaviors arise in situations where terrains are non-flat, disjointed, or extremely cluttered. The majority of bipedal locomotion frameworks have been historically focused on flat terrain or mildly rough terrain locomotion behaviors. Some of them are making their way into dynamically climbing stairs or inclined terrains. Additionally, Raibert accomplished hopping locomotion over rough terrains in the middle-80s. In contrast, our effort is centered around the goals of (1) providing metrics of robustness in rough terrain for robust control of the locomotion behaviors, (2) generalizing gaits to any types of surfaces including disjointed terrains, (3) providing formal tools to study planning, robustness, and reachability of the non-periodic gaits, and (4) demonstrating the ability of our framework to deal with large external disturbances.

Future work will focus on: (i) Experimental validations of the proposed planning and control strategy, where modeling, pose estimation, and kinematic errors, among other problems will greatly impact performance. (ii) Proposing a more realistic robot model that incorporates swing leg dynamics, and yaw and roll angular momentum. (iii) Extending the proposed framework for planning and control of omnidirectional dynamic locomotion in rough terrains. (iv) Proposing a realistic terrain perception model that does not assume perfect information about the terrain structure.

### Funding

This work was supported by the Office of Naval Research, ONR Grant [grant #N000141210663], NASA Johnson Space Center, NSF/NASA NRI Grant [grant #NNX12AM03G], and NSF CPS Synergy Grant [grant #1239136].

## Appendix

### A Index to Multimedia Extensions

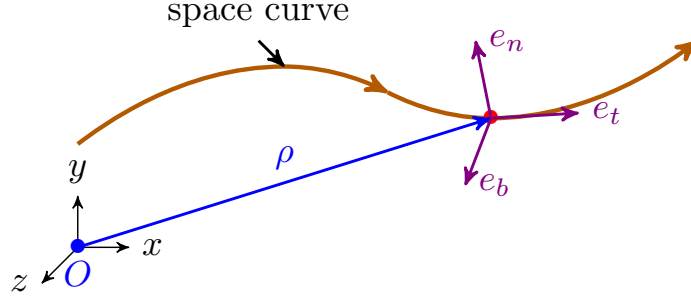
Extension	Type	Description
1	Video	Demonstrations of four locomotion simulations including (1) seven-step phase-space motion planning; (2) dynamic walking over rough terrains; (3) dynamic walking under external disturbances; (4) bouncing maneuver over disjointed terrain.

### B Phase-Space Manifold

The desired behavior of the outputs lie in manifolds  $\mathcal{M}_i$  as shown in Eq. (5). Here we present a brief review of space curves and surfaces that relate to the phase-space manifold (PSM) and present a Riemannian geometry metric that can be generalized to this family of problems.

The trajectory of the center of mass (CoM) of the robot is a space curve in 3D,  $\mathbf{p}_{\text{CoM}} = \{x, y, z\} \in \mathbb{R}^3$ . Also, for a particular output  $y_i$ , if we consider the case of an output-task with relative order  $r_i = 3$ , the manifold  $\mathcal{M}_i \in \mathbb{R}^{r_i}$  is a space curve  $\mathcal{C}_i$  in Euclidean three-dimensional space (see Fig. 16). We assume that the curve is parametrized by an arc-length parameter  $\zeta$  that we refer to as the progression variable. Hence the position vector  $\boldsymbol{\rho}_i$  of any point on the curve can be defined by specifying the value of  $\zeta$ ,

$$\boldsymbol{\rho}_i(\zeta) = \sum_{k=1}^{r_i} \xi_k(\zeta) \mathbf{E}_k = \xi_1(\zeta) \mathbf{E}_1 + \xi_2(\zeta) \mathbf{E}_2 + \xi_3(\zeta) \mathbf{E}_3. \quad (39)$$



**Fig. 16.** A space curve showing the evolution of the Frenet triad.

where,  $\mathbf{E}_k$  is the unit vector in the  $k$ -axis of the Euclidean space and  $\xi_k(\zeta)$  is the projection coordinate of  $\boldsymbol{\rho}$  on  $\mathbf{E}_k$ . A unit tangent vector  $\mathbf{e}_t$  to the curve can also be defined,

$$\mathbf{e}_t = \frac{\partial \boldsymbol{\rho}_i}{\partial \zeta}. \quad (40)$$

The derivative of this vector defines the curvature  $\kappa$  and the unit normal vector  $\mathbf{e}_n$ ,

$$\frac{\partial \mathbf{e}_t}{\partial \zeta} = \kappa \mathbf{e}_n, \quad \text{where } \kappa = \left\| \frac{\partial \mathbf{e}_t}{\partial \zeta} \right\|. \quad (41)$$

In the case of a space surface (where,  $\boldsymbol{\rho}_i$  belongs to a manifold  $\mathcal{M}_i$  in the output phase-space), instead of a vector  $\mathbf{e}_t$ , we have a tangent manifold, denoted by  $T_{\mathcal{M}_i}$ . The tangent space at any point can be mapped to the vector  $\boldsymbol{\chi}_i \in \mathbb{R}^{r_i-1}$  that spans  $T_{\mathcal{M}_i}$ . Without loss of generality, the actual motion is a specific line in space curve  $\mathcal{M}_i$ . The tangent vector in the manifold  $\mathcal{M}_i$  is  $\mathbf{e}_\zeta = (\mathbf{e}_i)_t$  while the cotangent vector in the manifold is  $\mathbf{e}_\sigma = (\mathbf{e}_i)_n$ .  $\mathbf{e}_\sigma$  denotes the normal deviation distance from the surface  $\sigma_i$ . For  $r_i = 3$ , the binormal vector,  $\mathbf{e}_b$  is orthogonal to  $\mathbf{e}_t$  and  $\mathbf{e}_n$ . These three vectors are called the Frenet space. These three Frenet frame vectors are proportional to the first three derivatives of the curve  $\boldsymbol{\rho}$ , as a benefit of taking the arc length  $\zeta$  as the parameter.

In disturbance-free cases, the system will remain in the manifold if it starts on it. It can be considered as the zero dynamics of the surface deviation,  $\sigma_i$ . When disturbance occurs, the state may escape the manifold and the controller should bring it back for recovery. To define a metric on the manifold itself and normal to it, we use Riemannian Geometry. In general, we treat each manifold  $\mathcal{M}_i$  in Eq. (5) of the task-space  $i^{\text{th}}$ -coordinate as independent from each other. The actual task manifold,  $\mathcal{M}$  is the intersection of all  $\mathcal{M}_i$  manifolds,

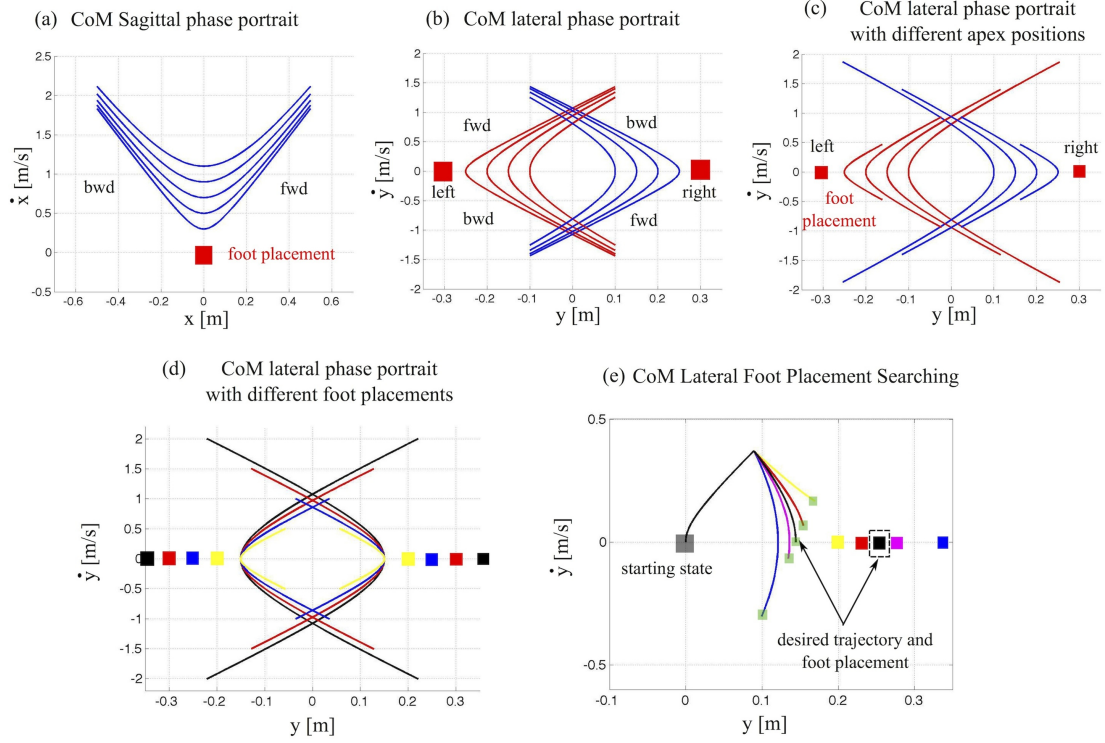
$$\mathcal{M} = \bigcap_i \mathcal{M}_i. \quad (42)$$

This manifold also has a tangent manifold  $T_{\mathcal{M}} \in \mathbb{R}^r$ , where  $r = \sum_i r_i$ . Each manifold  $\mathcal{M}_i$ , separately have a null-space (cotangent manifold,  $T_{\mathcal{M}_i}^*$ ) and their intersection is the task cotangent manifold,  $T_{\mathcal{M}}^*$ .

## C Numerical Integration

For the nonlinear Eq. (12), we assume that  $\ddot{x}$  is approximately constant for small increments of time. Since  $f$  is normally highly nonlinear, numerical integration algorithms are normally used. More details are explained in



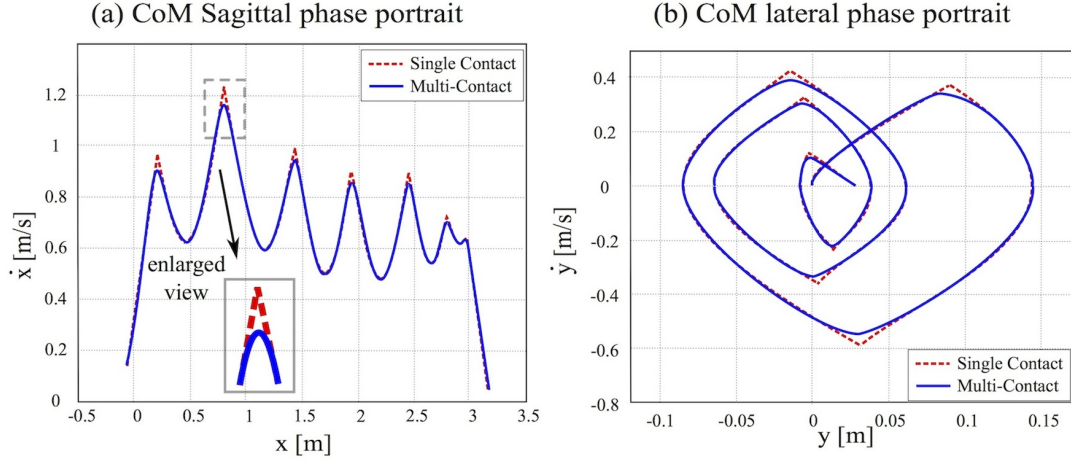


**Fig. 17.** Numerical integration of Sagittal and lateral dynamics. Numerical Integration is used for phase-space prediction of Sagittal and lateral prismatic inverted pendulum dynamics in Eq. (12). The phase portraits (a) and (b) correspond to the Sagittal and lateral CoM phase behaviors given desired foot contact locations (red boxes), a desired CoM surface of motion, and initial position and velocity conditions. If the same time duration is guaranteed for each trajectory, we can derive two different trajectories shown in (c) and (d) based on different initial conditions. (c) shows lateral CoM behaviors given a fixed lateral foot placement and varying initial lateral position conditions. (d) corresponds to CoM trajectories derived given varying lateral foot placements and a fixed initial condition. In (e), we analyze lateral CoM trajectories of two consecutive steps with varying lateral foot placements. As the foot placement moves further apart, the acceleration becomes larger and the CoM position transverses (at the Sagittal apex) less in the  $y$  direction.

[Sentis and Slovic (2011); Zhao (2013)]. The pipeline for finding state-space trajectories goes as follows: (1) choose a small time perturbation  $\epsilon$ , (2) given known velocities  $\dot{x}_k$  and accelerations  $\ddot{x}_k$ , we derive the next velocity  $\dot{x}_{k+1}$ , and the next position  $x_{k+1}$ , (3) plot the point  $(x_{k+1}, \dot{x}_{k+1})$  in the phase-plane. We can iterate this recursion both forward and backward. Based on this pipeline, one-step Sagittal and lateral manifold with different initial conditions are shown in Fig. 17.

## D Multi-Contact Maneuvers

The objective of this section is to incorporate multi-contact transitions into our gait planner to achieve more natural motions. Towards this goal, we fit a polynomial function with the desired smooth behavior to the phase curve in the vicinity of the step transition. For this process, desired boundary values of position, velocity and acceleration are given by the gait designer. It is necessary to also take into account time constraints to guarantee the synchronization of the Sagittal and lateral behaviors. Boundary and timing conditions allow us to calculate the coefficients of the polynomials. More mathematical details of this approach can be found in [Zhao and Sentis (2012)]. In our case, a



**Fig. 18.** Integration of multi-contact transition phases. The subfigures (a) and (b) are similar to their counterparts of Fig. 2 but with an additional multi-contact phase. By using 5<sup>th</sup> order polynomials and guaranteeing continuity with the existing curves, we get the polynomial parameters for curve fitting.

multi-contact phase is created to occupy 25% of the total time slot for a given step. This percentage is adjustable based on the desired walking profile. The results are illustrated in Fig. 18.

## E Proof of Phase-Space Tangent Manifold

In the nominal control case, the angular momentum  $\tau_y$  of Eq. (12) is zero. For this case, the Sagittal inverted pendulum dynamics are simple,  $\ddot{x} = \omega^2(x - p_x)$ , where  $\omega$  is constant for a given step. Since the foot placement  $p_x$  is also constant over the step, then  $\ddot{p}_x = \dot{p}_x = 0$ . Therefore the previous equation is equivalent to  $\ddot{x} - \ddot{p}_x = \omega^2(x - p_x)$ . Let us define a transformation  $\tilde{x} = x - p_x$ . We can then write  $\ddot{\tilde{x}} = \omega^2\tilde{x}$ . Using Laplace transformations, we have  $s^2\tilde{x}(s) - \tilde{x}_0 - s\dot{\tilde{x}}_0 = \omega^2\tilde{x}(s)$ . Based on this, we get

$$\tilde{x}(t) = \mathcal{L}^{-1}\left\{\frac{\tilde{x}_0 + s\dot{\tilde{x}}_0}{s^2 - \omega^2}\right\}. \quad (43)$$

Solving the equation above, we can derive an analytical solution

$$\tilde{x}(t) = \frac{\tilde{x}_0(e^{\omega t} + e^{-\omega t})}{2} + \frac{\dot{\tilde{x}}_0(e^{\omega t} - e^{-\omega t})}{2\omega} = \tilde{x}_0 \cosh(\omega t) + \frac{1}{\omega} \dot{\tilde{x}}_0 \sinh(\omega t), \quad (44)$$

and by taking its derivative, we get

$$\dot{\tilde{x}}(t) = \omega \tilde{x}_0 \sinh(\omega t) + \dot{\tilde{x}}_0 \cosh(\omega t). \quad (45)$$

These two equations can be further expressed as

$$x(t) = (x_0 - p_x) \cosh(\omega t) + \frac{1}{\omega} \dot{x}_0 \sinh(\omega t) + p_x \quad (46)$$

$$\dot{x}(t) = \omega(x_0 - p_x) \sinh(\omega t) + \dot{x}_0 \cosh(\omega t) \quad (47)$$



Now we have the following state space formulation

$$\begin{pmatrix} x(t) - p_x \\ \dot{x}(t) \end{pmatrix} = \begin{pmatrix} x_0 - p_x & \dot{x}_0/\omega \\ \dot{x}_0 & \omega(x_0 - p_x) \end{pmatrix} \begin{pmatrix} \cosh(\omega t) \\ \sinh(\omega t) \end{pmatrix}$$

which implies

$$\begin{pmatrix} \cosh(\omega t) \\ \sinh(\omega t) \end{pmatrix} = \frac{1}{\omega(x_0 - p_x)^2 - \dot{x}_0^2/\omega} \begin{pmatrix} \omega(x_0 - p_x) & -\dot{x}_0/\omega \\ -\dot{x}_0 & x_0 - p_x \end{pmatrix} \begin{pmatrix} x - p_x \\ \dot{x} \end{pmatrix} \quad (48)$$

since  $\cosh^2(x) - \sinh^2(x) = 1$ , we get

$$(\omega(x_0 - p_x)(x - p_x) - \dot{x}_0\dot{x}/\omega)^2 - (-\dot{x}_0(x - p_x) + \dot{x}(x_0 - p_x))^2 = (\omega(x_0 - p_x)^2 - \dot{x}_0^2/\omega)^2$$

After expanding the square terms and moving all terms to one side, we obtain

$$(x_0 - p_x)^2(2\dot{x}_0^2 - \dot{x}^2 + \omega^2(x - x_0)(x + x_0 - 2p_x)) - \dot{x}_0^2(x - p_x)^2 + \dot{x}_0^2(\dot{x}^2 - \dot{x}_0^2)/\omega^2 = 0$$

which is the phase-space tangent manifold  $\sigma$  defined in Proposition 1.  $\square$

## F Proof of Phase-Space Cotangent Manifold

In this case, we use the tangent manifold in Eq. (23) to derive the cotangent manifold  $\zeta$ . By taking the derivative of Eq. (23), we have

$$d\sigma = \frac{\partial\sigma}{\partial x}dx + \frac{\partial\sigma}{\partial\dot{x}}d\dot{x},$$

where

$$\frac{\partial\sigma}{\partial x} = -2\dot{x}_{\text{apex}}^2(x - x_{\text{foot}}), \quad \frac{\partial\sigma}{\partial\dot{x}} = 2\dot{x}_{\text{apex}}^2\dot{x}/\omega^2.$$

The  $\sigma$  manifold's normal vector is given by its gradient,  $\mathbf{e}_n = (-2\dot{x}_{\text{apex}}^2(x - x_{\text{foot}}), 2\dot{x}_{\text{apex}}^2\dot{x}/\omega^2)^T$ , and its tangent vector is orthogonal to  $\mathbf{e}_n$ , i.e.,  $\mathbf{e}_t = (2\dot{x}_{\text{apex}}^2\dot{x}/\omega^2, 2\dot{x}_{\text{apex}}^2(x - x_{\text{foot}}))^T$ . Since  $\zeta$  is orthogonal to  $\sigma$ , the tangent vector of  $\zeta$  is the normal vector of  $\sigma$ , i.e.,

$$d\zeta = \frac{\partial\zeta}{\partial x}dx + \frac{\partial\zeta}{\partial\dot{x}}d\dot{x},$$

where

$$\frac{\partial\zeta}{\partial x} = 2\dot{x}_{\text{apex}}^2\dot{x}/\omega^2, \quad \frac{\partial\zeta}{\partial\dot{x}} = -2\dot{x}_{\text{apex}}^2(x - x_{\text{foot}})$$

Via the equations above, we can further obtain

$$\frac{d\dot{x}}{dx} = -\frac{\dot{x}}{\omega^2(x - x_{\text{foot}})} \quad \Rightarrow \quad \omega^2 \int_{\dot{x}_0}^{\dot{x}} \frac{d\dot{x}}{\dot{x}} = - \int_{x_0}^x \frac{dx}{x - x_{\text{foot}}}$$

---

**Algorithm 3** Overall Hybrid Robust Locomotion Planning Structure

---

```
1: Initialize walking step index  $k \leftarrow 1$ , discrete state  $q$ , initial condition  $\mathcal{I}_q, \epsilon$  for invariant bundle  $\mathcal{B}_q(\epsilon)$ , stage
   update indicator  $b_{\text{update}} \leftarrow \text{false}$ .
2: while  $x_q \notin \mathcal{G}_a^{[\delta_j]}(q, q+1)$  do
3:   if  $x_q \in \mathcal{G}_d^{[\delta_j]}(q, q_{\text{dist}})$  then
4:     Execute  $\Delta_d^{[\delta_j]}(q, q_{\text{dist}}, x_{q_{\text{dist}}})$  and quantize disturbed state  $(x_q)_{\text{dist}}$ .
5:     Generate optimal policies  $(\zeta, \dot{x}, \ddot{x}, \tau, \omega, \mathcal{L}, \mathcal{V})_{\text{opt}}$  by dynamic programming.
6:      $b_{\text{update}} \leftarrow \text{true}$ .
7:     Compute the phase-space manifold  $\sigma_{\text{trans}}$  by Eq. (22) at transition phase.
8:     if  $(x_q)_{\text{trans}} \notin \mathcal{R}_q$  then
9:       Re-plan  $x_{\text{foot}_{q+1}}$  by Eq. (37) and search  $y_{\text{foot}_{q+1}}$  by Algorithm 2.
10:    end if
11:  end if
12:  Compute  $\sigma_{i+1}$  over domain  $\mathcal{D}_q$  by Eq. (22).
13:  if  $b_{\text{update}}$  is true then
14:    Update stage index  $i_{\text{stage}}$  of recovery optimal control inputs.
15:  end if
16:  if  $x_q \notin \mathcal{B}_q$  then
17:    Compute  $u_{c_{i+1}} =: (\tau_y, \omega)_{i_{\text{stage}}}$  by Eq. (28a) and assign  $\ddot{x}_{i+1} \leftarrow \ddot{x}_{\text{opt}}(i_{\text{stage}})$ .
18:  else
19:    Compute  $u_{c_{i+1}} =: (\tau_y, \omega)_{i+1}$  by Eq. (28b) and assign  $\ddot{x}_{i+1}$  by Eq. (12).
20:  end if
21:  Evolve  $(x_{i+1}, \dot{x}_{i+1})$  over domain  $\mathcal{D}_q$  numerically.
22:   $i \leftarrow i + 1$ .
23: end while
24:  $q \leftarrow q + 1$ , re-assign  $\mathcal{I}_{q+1}, b_{\text{update}} \leftarrow \text{false}$  and jump to line 2 for next walking step.
```

---

then we have

$$\ln\left(\frac{\dot{x}}{\dot{x}_0}\right)^{\omega^2} + \ln \frac{x - x_{\text{foot}}}{x_0 - x_{\text{foot}}} = 0 \quad \Rightarrow \quad \left(\frac{\dot{x}}{\dot{x}_0}\right)^{\omega^2} \frac{x - x_{\text{foot}}}{x_0 - x_{\text{foot}}} = 1$$

Thus, the cotangent manifold can be defined as

$$\zeta = \zeta_0 \left(\frac{\dot{x}}{\dot{x}_0}\right)^{\omega^2} \frac{x - x_{\text{foot}}}{x_0 - x_{\text{foot}}}$$

where the constant  $\zeta_0$  is a nonnegative scaling factor.  $(x_0, \dot{x}_0)$  is the initial condition at  $\zeta = \zeta_0$ . The equation above is the phase-space cotangent manifold  $\zeta$  defined in Proposition 2.  $\square$

## References

- Abdallah, M. and A. Goswami (2005). A biomechanically motivated two-phase strategy for biped upright balance control. In *IEEE-RAS International Conference on Robotics and Automation, 2005*, pp. 1996–2001.
- Arslan, O. and U. Saranlı (2012). Reactive planning and control of planar spring–mass running on rough terrain. *IEEE Transactions on Robotics* 28(3), 567–579.
- Audren, H., J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida (2014). Model preview control in multi-contact

- motion-application to a humanoid robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4030–4035.
- Branicky, M. S., V. S. Borkar, and S. K. Mitter (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions Automatic Control* 43(1), 31–45.
- Byl, K. and R. Tedrake (2009). Metastable walking machines. *The International Journal of Robotics Research* 28(8), 1040–1064.
- Dai, H. and R. Tedrake (2012). Optimizing robust limit cycles for legged locomotion on unknown terrain. In *IEEE Conference on Control and Decision*, pp. 1207–1213.
- Englsberger, J., C. Ott, and A. Albu-Schaffer (2015). Three-dimensional bipedal walking control based on divergent component of motion. *IEEE Transactions on Robotics* 31(2), 355–368.
- Feng, S., X. Xinjilefu, C. G. Atkeson, and J. Kim (2015). Optimization based controller design and implementation for the atlas robot in the darpa robotics challenge finals. In *IEEE-RAS International Conference on Humanoid Robots, 2015*, pp. 1028–1035.
- Fernández-Rodríguez, B. (1988). *Control of Multivariable Nonlinear Systems by the Sliding Mode Method*. Ph. D. thesis, Massachusetts Institute of Technology.
- Frazzoli, E. (2001). *Robust hybrid control for autonomous vehicle motion planning*. Ph. D. thesis, Massachusetts Institute of Technology.
- Grizzle, J. W., C. Chevallereau, R. W. Sinnet, and A. D. Ames (2014). Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica* 50(8), 1955–1988.
- Hamed, K. A., B. G. Buss, and J. W. Grizzle (2015). Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations. In *The International Journal of Robotics Research*, in press.
- Hauser, K. (2014). Fast interpolation and time-optimization with contact. *The International Journal of Robotics Research* 33(9), 1231–1250.
- Hobbelen, D. G. and M. Wisse (2007). A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm. *IEEE Transactions on Robotics* 23(6), 1213–1224.
- Hof, A. L. (2008). The ‘extrapolated center of mass’ concept suggests a simple control of balance in walking. *Human Movement Science* 27(1), 112–125.
- Hofmann, A. (2006). *Robust execution of bipedal walking tasks from biomechanical principles*. Ph. D. thesis, Massachusetts Institute of Technology.
- Hyon, S. and G. Cheng (2007). Disturbance rejection for biped humanoids. In *IEEE-RAS International Conference on Robotics and Automation, 2007*, pp. 2668–2675.
- Isidori, A. (1985). Nonlinear control systems: An introduction. In M. Thomas (Ed.), *Lecture Notes in Control and Information Sciences*. Berlin: Springer-Verlag.
- Komura, T., H. Leung, S. Kudoh, and J. Kuffner (2005). A feedback controller for biped humanoids that can counteract large perturbations during gait. In *IEEE-RAS International Conference on Robotics and Automation, 2005*, pp. 1989–1995.
- Kuindersma, S., R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake (2015). Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 1–27.
- Kuo, A. D. and F. E. Zajac (1992). Human standing posture: multi-joint movement strategies based on biomechanical constraints. *Progress in Brain Research* 97, 349–358.
- Lygeros, J., C. Tomlin, and S. Sastry (2008). *Hybrid systems: modeling, analysis and control*. Preprint.

- Majumdar, A. (2013). *Robust online motion planning with reachable sets*. Master thesis, Massachusetts Institute of Technology.
- Manchester, I. R., U. Mettin, F. Iida, and R. Tedrake (2011). Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research* 30(3), 265–279.
- Mordatch, I., M. De Lasa, and A. Hertzmann (2010). Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics* 29(4), 71.
- Mordatch, I., E. Todorov, and Z. Popović (2012). Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics* 31(4), 43.
- Morisawa, M., S. Kajita, F. Kanehiro, K. Kaneko, K. Miura, and K. Yokoi (2012). Balance control based on capture point error compensation for biped walking on uneven terrain. In *IEEE-RAS International Conference on Humanoid Robots, 2012*, pp. 734–740.
- Nguyen, Q. and K. Sreenath (2015). Optimal robust control for bipedal robots through control lyapunov function based quadratic programs. In *Robotics: Science and Systems*.
- Pham, Q.-C., S. Caron, and Y. Nakamura (2013). Kinodynamic planning in the configuration space via admissible velocity propagation. In *Robotics: Science and Systems*.
- Pratt, J., J. Carff, S. Drakunov, and A. Goswami (2006). Capture point: A step toward humanoid push recovery. In *IEEE-RAS International Conference on Humanoid Robots, 2006*, pp. 200–207.
- Pratt, J., C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt (2001). Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research* 20(2), 129–143.
- Raibert, M. H. (1986). *Legged robots that balance*. MIT press.
- Ramezani, A., J. W. Hurst, K. A. Hamed, and J. Grizzle (2014). Performance analysis and feedback control of atrias, a three-dimensional bipedal robot. *Journal of Dynamic Systems, Measurement, and Control* 136(2), 021012.
- Saglam, C. O. and K. Byl (2014). Robust policies via meshing for metastable rough terrain walking. In *Robotics: Science and Systems, 2014*.
- Schouwenaars, T., B. Mettler, E. Feron, and J. P. How (2003). Robust motion planning using a maneuver automation with built-in uncertainties. In *American Control Conference, 2003*, pp. 2211–2216.
- Sentis, L., J. Park, and O. Khatib (2010). Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Transactions on Robotics* 26(3), 483–501.
- Sentis, L. and M. Slovich (2011). Motion planning of extreme locomotion maneuvers using multi-contact dynamics and numerical integration. In *IEEE-RAS International Conference on Humanoid Robots, 2011*, pp. 760–767.
- Srinivasan, M. and A. Ruina (2006). Computer optimization of a minimal biped model discovers walking and running. *Nature* 439(7072), 72–75.
- Stephens, B. (2007). Humanoid push recovery. In *IEEE-RAS International Conference on Humanoid Robots, 2007*, pp. 589–595.
- Stephens, B. J. and C. G. Atkeson (2010). Push recovery by stepping for humanoid robots with force controlled joints. In *IEEE-RAS International Conference on Humanoid Robots, 2010*, pp. 52–59.
- Takenaka, T., T. Matsumoto, and T. Yoshiike (2009). Real time motion generation and control for biped robot-1st report: Walking gait pattern generation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009*, pp. 1084–1091.
- Tassa, Y., T. Erez, and E. Todorov (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012*, pp. 4906–4913.
- Utkin, V. I. (2013). *Sliding modes in control and optimization*. Springer Science & Business Media.
- Wieber, P.-B. (2006). Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *IEEE-RAS International Conference on Humanoid Robots, 2006*, pp. 137–142.

- Zhao, J., S. Schutz, and K. Berns (2013). Biologically motivated push recovery strategies for a 3d bipedal robot walking in complex environments. In *IEEE International Conference on Robotics and Biomimetics, 2013*, pp. 1258–1263.
- Zhao, Y. (2013). *Phase Space Planning for Robust Locomotion*. Master thesis, The University of Texas at Austin.
- Zhao, Y., D. Kim, G. Thomas, and L. Sentis (2015). Hybrid multi-contact dynamics for wedge jumping locomotion behaviors. In *ACM International Conference on Hybrid Systems: Computation and Control, 2015*.
- Zhao, Y. and L. Sentis (2012). A three dimensional foot placement planner for locomotion in very rough terrains. In *IEEE-RAS International Conference on Humanoid Robots, 2012*, pp. 726–733.
- Zhou, K., J. C. Doyle, K. Glover, et al. (1996). *Robust and optimal control*. Prentice hall New Jersey.